An ORSAC method for data cleaning inspired by RANSAC

Thomas Jenkins, Autumn Goodwin, Sameerah Talafha
Computer Vision Research, Vectech, Baltimore, United States

Article Info

Article history:

Received Jan 11, 2024
Revised May 23, 2024
Accepted Jun 18, 2024

Keywords:

Classification
Computer vision
Data cleaning
Label noise
RANSAC

ABSTRACT

In classification problems, mislabeled data can have a dramatic effect on the capability of a trained model. The traditional method of dealing with mislabeled data is through expert review. However, this is not always ideal, due to the large volume of data in many classification datasets, such as image datasets supporting deep learning models, and the limited availability of human experts for reviewing the data. Herein, we propose an ordered sample consensus (ORSAC) method to support data cleaning by flagging mislabeled data. This method is inspired by the random sample consensus (RANSAC) method for outlier detection. In short, the method involves iteratively training and testing a model on different splits of the dataset, recording misclassifications, and flagging data that is frequently misclassified as probably mislabeled. We evaluate the method by purposefully mislabeling subsets of data and assessing the method's capability to find such data. We demonstrate with three datasets, a mosquito image dataset, CIFAR-10, and CIFAR-100, that this method is reliable in finding mislabeled data with a high degree of accuracy. Our experimental results indicate a high proficiency of our methodology in identifying mislabeled data across these diverse datasets, with performance assessed using different mislabeling frequencies.

Corresponding Author:

Thomas Jenkins
Computer Vision Research, Vectech
Baltimore, Maryland, United States
Email: thomas@vectech.io

1. INTRODUCTION

Mislabeled data, or label noise, is a common issue in classification problems and can significantly compromise the efficiency of models designed for classification tasks. Traditional methods often rely on expert review of the dataset, which may not be practical due to the substantial volume of data in many classification datasets and the limited availability of human expertise for review. This issue is problematic in various applications of machine learning, especially in computer vision. It becomes particularly challenging for extremely fine-grained datasets where few human experts are available for labeling the data [1]-[3]. In these cases, automated or semi-automated methods to identify mislabeled data for exclusion, or to flag subsets of the data for review, are highly desirable. When flagging subsets of the data for review, low precision may be acceptable if there is very high sensitivity, since the method may serve as a screening tool rather than as the final decision maker on a sample.

The ubiquitous issue of data mislabeling not only hinders the accuracy of machine learning models but also stands as a roadblock to the advancement of sophisticated, fine-grained classification tasks. The impact of mislabeled data is even more pronounced in datasets where labeling requires specialized expertise, such as in medical diagnostics [4] or species identification in biological sciences [5]-[7]. Current methodologies typically involve manual or semi-automated processes of data review, which are time-consuming and may not be feasible due to the scarcity of domain experts. This scarcity is especially evident in the shortage of taxonomists, contributing to the global taxonomic impediment [8]-[10]. Even the slightest misclassification can have far-reaching implications in fields like medical imaging and environmental conservation, underlining the critical need for a more efficient and reliable approach to addressing data mislabeling. This need is doubly important, as the shortage of experts not only necessitates greater reliance on computer-aided decision-making but also impedes the development of datasets to support classification models, given the extreme scarcity of an expert's time.

Recent advancements in machine learning, particularly in deep learning, have opened new avenues for addressing the issue of mislabeled data. Studies have demonstrated that deep learning models can be trained not only to perform classification tasks but also to identify inconsistencies in labeling by learning complex patterns and anomalies in the data [11], [12]. Moreover, the integration of active learning and human-in-the-loop approaches has shown potential in efficiently managing large datasets by selectively querying human input on the most ambiguous cases [13]. This combination of automated learning and selective expert involvement presents a promising direction for tackling the challenge of label noise in datasets. Incorporating these advancements, our proposed ordered sample consensus (ORSAC) method aligns with this emerging trend. It leverages the strengths of deep learning for the initial flagging of potential mislabels and allows for targeted expert review, thus optimizing the use of limited expert resources. This approach not only augments the efficiency of data cleaning but also contributes to the development of more robust and accurate classification models, particularly in specialized fields where expert knowledge is vital yet scarce.

In this paper, we introduce the ORSAC method to tackle the persistent challenge of data mislabeling in classification tasks. Our approach employs an iterative process of training and testing a model on different portions of the dataset, documenting misclassifications, and marking frequently misclassified data as potential mislabels. The intuition behind this process is that a model trained on mostly correct data will likely misclassify mislabeled data found in the test set [14]. While inspired by the established random sample consensus (RANSAC) technique, ORSAC is specifically adapted to address the complex issue of label noise in datasets. This adaptation allows for more accurate identification of mislabeled data, which is particularly useful in intricate, fine-grained datasets. At the same time, it retains the essential scalability needed for handling large volumes of data. Consequently, ORSAC represents a meaningful advancement in the area of data quality for machine learning. It effectively combines algorithmic precision with the practical necessity of scalability, offering a valuable tool in the ongoing effort to improve data-cleaning processes. While ORSAC builds upon existing methodologies, its specific focus on label noise and its scalable approach make it a noteworthy contribution to the field, particularly for machine learning applications dealing with large and complex datasets.

To validate our approach, we intentionally mislabel certain subsets of data and assess our method's ability to identify such data. We utilize three distinct datasets: a mosquito image dataset, CIFAR-10, and CIFAR-100. Our experimental results demonstrate a high proficiency of our methodology in identifying mislabeled data across these diverse datasets, with performance documented under various mislabeling frequencies. Given these results, we advocate for the use of ORSAC as an optimal process for routine data cleaning, especially to support the growth of fine-grained classification datasets where data cleanliness is of utmost importance, but expert labeler time is a limited resource.

The remainder of this paper is delves into the motivations behind our research, drawing on established theories and past studies that have contributed to the development of our innovative methodology and approach. Section 2 provides a review of existing literature pertinent to our study. In section 3., we elaborate on our research design and analytical approach. We detail the iterative classification method for obtaining a misclassification frequency. In sub-section 4.1. discusses our methods for assessing the effectiveness of the methodology, and in sub-section 4.2. outlines our experimental setup, with the results presented and discussed in 5.1. Subsequently, in sub-section 5.2. provides a thorough analysis of these results and offers a nuanced discussion of our findings. We acknowledge the limitations of our study while also emphasizing its strengths and identifying potential areas for future research, thereby linking our work to the broader trajectory of this field.

The problem of mislabeled data detection is similar in nature to the problem of outlier detection. Both outliers and mislabeled data are non-representative of the class they are labeled as. RANSAC is an algorithm commonly used to train a regression model in the presence of outliers. It works by first training the regression model on the minimum amount of data needed to fit the model, then determining the number of inliers that are within a threshold of the fitted model. This process is repeated a certain number of iterations, after which the model with the most inliers is chosen. The model can also be retrained using the inliers defined by the best model. ORSAC draws inspiration from the traditional RANSAC algorithm in that it trains several iterations of the model, each time using different portions of the dataset as the training set. It differs in that it is designed to function with a classification model as opposed to a regression model, and the final product is not intended to be a trained model, but rather a set of potentially mislabeled data points. These data points may be excluded from the dataset, though this may have unintended consequences, potentially removing important variability from a complex dataset, resulting in a model less robust to real-world variability. For this reason, although we assess accuracy improvements from the simple exclusion of the data, we recommend an ideal process of flagging these potentially mislabeled data points for expert review and final decision-making.

The adaptation of outlier detection principles to the realm of classification models, particularly in the context of image datasets, heralds new opportunities for enhancing data integrity in machine learning [15]. This conceptual shift, embodied by ORSAC, underscores a growing appreciation within the field: methodologies honed for one type of model, such as regression, can be adapted to offer significant insights and solutions for others, like image classification tasks [16], [17]. Such cross-pollination of techniques not only diversifies our toolkit for data preprocessing but also fosters a more comprehensive understanding of data quality in various machine learning models, especially those dealing with fine-grained, image-based datasets [18], [19].

ORSAC's nuanced approach, which involves flagging potential mislabels for expert review rather than outright exclusion, demonstrates a deep understanding of the complexities and subtleties present in image data. Compared to the incorrectly labeled samples we aim to exclude, the correctly labeled outliers are sometimes the most important data to include to manage data sampling biases. This method ensures the preservation of the richness and intricacies inherent in these datasets, which is essential for developing models that are robust and effective in interpreting the nuanced visual information typical of real-world scenarios. By integrating these nuanced approaches, ORSAC contributes to a more dynamic and effective strategy for handling data quality challenges in image-based machine learning.

2. RELATED WORKS

Addressing the pervasive issue of data mislabeling has been the focus of extensive research over the years, with numerous methodologies proposed to tackle this challenge. The complexity of this problem, heightened by the diversity of datasets and the significant consequences of mislabeling, has catalyzed the creation of a range of innovative solutions. These solutions have evolved from traditional, manual techniques to more advanced computational methods, reflecting the dynamic nature of research in this area. This evolution has been driven by various factors, including the need for greater scalability, the advent of sophisticated algorithmic approaches, and the increasing complexity of datasets, tracing their development from early statistical methods to contemporary machine learning techniques [20]-[22]. In this section, we detail some of the noteworthy methods proposed in the literature, each with its unique approach, advantages, and limitations.

The quintessential application of the RANSAC algorithm in limited data scenarios is exemplified in the domain of computer vision, specifically for tasks such as geometric shape detection, including line or plane fitting in 2D or 3D spaces. In a typical scenario involving a small dataset with spatial data points, some of which outline a geometric shape (e.g., a line in 2D) amidst random noise or outliers, RANSAC's stochastic approach becomes pivotal. It begins by randomly selecting a minimal subset of data points – just enough to define a geometric entity, like two points for a line in 2D space. A line is then fitted to these points, and the distances of all remaining data points to this model are computed. Those within a certain threshold are classified as inliers, refining the initial model estimation through multiple iterations, each involving the selection of a new random subset and accruing inliers. The model with the maximal aggregation of inliers is ultimately adopted as the representative of the underlying data structure. This methodology, particularly beneficial in scenarios with small datasets, mitigates the skew caused by outliers, leading to more accurate model representations [23]-[26].

Originally developed for applications in image analysis and automated cartography, RANSAC provided a robust solution for fitting models in data characterized by high proportions of noise or anomalies. Its capability to distinguish inliers from outliers without prior knowledge of data validity established it as a pioneering tool in computer vision and image processing. This foundational algorithm has significantly influenced the development of numerous subsequent methods in outlier detection and robust estimation across various fields, underscoring its enduring impact and versatility [27]-[29].

Building upon the foundational principles established by the RANSAC algorithm, several researchers have proposed innovative adaptations for tackling mislabeling in image datasets. Notably, Debnath *et al.* [17]

extended RANSAC's framework by incorporating a CNN trained on the ImageNet (ILSVRC2012) dataset to generate feature vectors. They then iteratively trained a support vector machine (SVM) to differentiate between outliers and inliers, evaluating the complete dataset in each iteration [17]. This method showcases the integration of deep learning with traditional algorithms to enhance outlier detection.

Expanding on this theme of combining methodologies, Moura *et al.* [30] introduced an approach using an ensemble of classifiers. Their method, trained on clean data, focuses on identifying mislabeled data within the test set through a voting system implemented with various threshold settings. This technique highlights the application of ensemble learning in ensuring data integrity.

Furthering this exploration, Feng *et al.* [31] tested the ensemble classifier approach in scenarios where mislabeling is present not only in the test set but also in the training set. They employed both majority and consensus thresholds to enhance the robustness of mislabel detection. This research underscores the importance of considering label noise in both the training and testing phases for more realistic data conditions.

In a different vein, Wu *et al.* [32] explored the concept of 'A Topological Filter for Learning with Label Noise.' Their method leverages the latent spatial distribution patterns of a noisy classifier trained on the dataset. By forming clusters of what is deemed to be clean data, their approach focuses on filtering out outliers, thus refining the dataset quality. This method exemplifies the utilization of topological data analysis in the context of label noise.

Additional methodologies in the realm of data mislabeling, such as those proposed by Ghorbani and Zou [33], and Koh *et al.* [34], focus on generating scores for each data point to identify label noise. Ghorbani and Zou utilize a method based on "shapley values," derived through a leave-one-out approach, where the lowest valued data are earmarked for review by trusted experts to detect label noise. On the other hand, Koh *et al.* [34] adopt a distinct strategy in score generation and interpretation. They leverage influence functions, and their approximations, to assess how significantly each training data point influences a model's parameters, with the premise that data with the highest influence scores are likely to contain label noise. However, the practical applications of both methods are somewhat constrained by the complexity of the problem they address. A notable limitation is their conceptual approach to identifying mislabeling, particularly in scenarios characterized by high inter-class similarity and notable intra-class variations, a defining feature of fine-grained problems. These methods presuppose that mislabeled data will exert a substantial impact on the model, affecting either its performance or its parameters in a pronounced manner. Such an assumption may not hold true in cases where the defining features of the data are less influential at a macro level of the model. In essence, if the features critical for classification are subtle or nuanced, these methodologies might not effectively identify the mislabeled data points.

Nguyen *et al.* [35] introduce a novel approach that contrasts with traditional data cleaning methods. Their strategy is integrated directly into the model's training loop, aiming to enhance the model's resilience to noisy labels without necessitating a separate data cleaning step. This innovative method, termed 'self-ensembling-predictions' or SELF, leverages the outputs of a single model across different training epochs. It involves recording the model's predictions on the training set at various epochs to establish a moving average of predictions for each data sample. This moving average is then compared to the sample's actual label. A discrepancy between the label and the moving average is interpreted as an indication of mislabeling, prompting the removal of the sample from the supervised training set in subsequent iterations. Notably, Nguyen *et al.* [35] method maintains engagement with the noisy labels by incorporating an unsupervised loss component in the training process. This aspect of their approach allows for continued learning from potentially mislabeled data, thereby balancing label correction with the retention of valuable data information. Many methods such as those proposed in Zhou *et al.* [36] and Park *et al.* [37] explicitly aim to relabel instances, the former using ensemble methods and the latter using a data pruning algorithm to find the subset of the dataset for which a predicted relabel is most likely correct. Other methods take the approach of making a model that is robust to label noise, as opposed to one that can identify instances of label noise [38], [39].

An essential aspect of the label noise problem lies in the complexity of the dataset under consideration. While many of the methods discussed earlier prove effective for datasets with clearly distinguishable classes, challenges arise when dealing with datasets where class differentiation is not as apparent, and critical features are subtle. This scenario is typical in what are known as fine-grained datasets, where even expert labelers and machine learning models may struggle to discern between classes due to the nuanced nature of the data [1]. In response to this specific challenge, ORSAC emerges as a tailored solution. It approaches these fine-grained, complex datasets by employing an iterative process with a sophisticated model. The iterative nature of ORSAC allows for a more nuanced understanding and identification of label noise, adapting to the subtle distinctions within the data. This methodical approach aims to enhance the accuracy of label noise detection in scenarios where traditional methods might falter due to the intricacy of the dataset.

In the context of ORSAC, there is substantial opportunity for further refinement, especially in improving its efficiency and precision. One promising avenue is leveraging GPU-specific libraries for parallel processing, which has the potential to significantly expedite computational tasks [40]-[42]. Such enhancements would make ORSAC more adept at handling larger and more complex datasets. A particularly exciting application of ORSAC lies in the field of biological and entomological research, where it can be used to analyze fine-grained images, such as those of mosquitoes, ticks, bees, and other small organisms. The capability of ORSAC to accurately differentiate subtle features in these images can be crucial for species identification and understanding biological diversity. Furthermore, the adaptation of ORSAC to these specific domains, combined with its iterative approach, offers a robust tool for researchers dealing with vast amounts of image data. These advancements not only demonstrate the adaptability and applicability of ORSAC in specialized fields but also underscore the need for continued research and development in advanced label noise correction methods, particularly for fine-grained image analysis.

3. PROPOSED METHOD

3.1. Method overview

As depicted in Figure 1, our ORSAC process encompasses four primary steps. Initially, the complete dataset, denoted as X, is partitioned into temporary subsets for training X^i_{Train}, validation X^i_{Valid}, and testing X^i_{Test}. In the first step, a classification model is trained on X^i_{Train} utilizing a fixed set of hyperparameters. The most efficacious model is chosen based on its performance on X^i_{Valid}. The second step involves evaluating this trained model using X^i_{Test} to generate predictions. In the third step, these predictions are compared with the actual labels. Here, the misclassification frequency, f, for each image in the test set is updated. The fourth step iterates the dataset splits using a sliding window technique, ensuring each sample is included in the test set an equal number of times. This procedure is repeated for n iterations. Post these iterations, the misclassification frequencies are compared to a predefined threshold, t. Images with frequencies meeting or exceeding t are flagged. These flagged images can either be eliminated from the dataset or subjected to manual review for potential exclusion or reclassification. This results in a refined dataset, $X^{'}$.

ORSAC attempts to flag samples which are likely to have been mislabeled in image classification tasks. First there is an initial data split. Then the method enters a loop for n iterations, i: 1) a CNN is trained on the training data, storing the best model of all epochs as evaluated by the validation set; 2) the trained model is evaluated on the test set; 3) update the misclassification frequency, f, by comparing the predictions of each image to its True label over each iteration; 4) iterate the data split using a sliding window approach. If i is more than n, exit the loop between steps 3 and 4. Then compare f to a previously set threshold t for each sample. If f is more than t, then the sample is flagged for either removal or review by a human expert

3.2. Iterative model training and misclassification frequency generation

Initially, the complete dataset X is segmented into temporary subsets for training X^i_{Train}, validation X^i_{Valid}, and testing X^i_{Test}, where i represents the current iteration, and n signifies the total number of iterations. A classification model, M, is trained on the X^i_{Train} subset using a consistent set of hyperparameters. The optimally trained model, M^*, is identified based on its performance on the X^i_{Valid} subset. This process is encapsulated by (1):

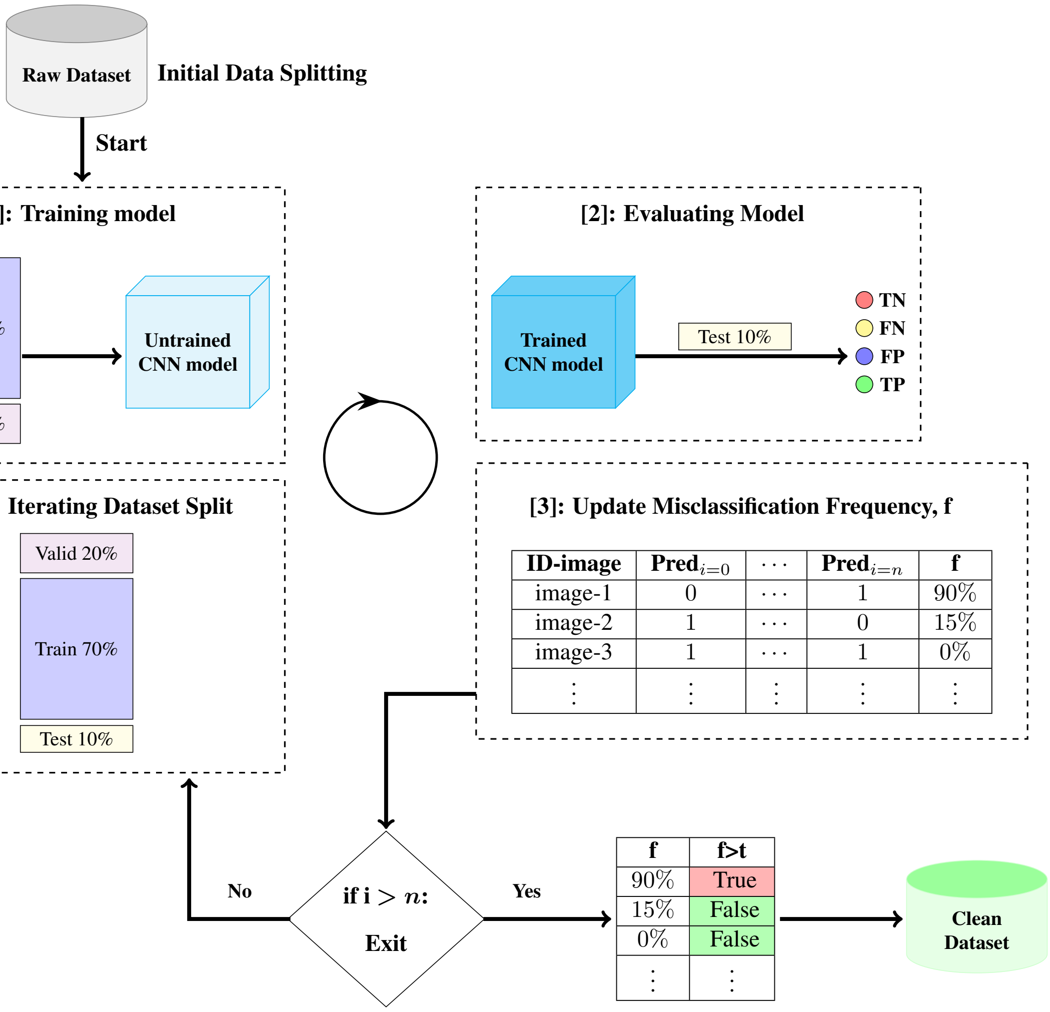
$$M^* = Train(M, X^i_{Train}, X^i_{Valid}) \tag{1}$$

upon training the model M^*, it is utilized to predict labels, y^i_{preds}, on the X^i_{Test} subset, as depicted by (2).

$$y^i_{preds} = Test(M^*, X^i_{Test}) \tag{2}$$

These y^i_{preds} are retained for subsequent comparison with the actual labels, y^i_{True}, to identify misclassifications. A sliding window technique is implemented to allocate the next split of the dataset $(X^i_{Train}, X^i_{Valid}, X^i_{Test})$, ensuring equitable inclusion of all samples in the test set across iterations. Following the classification phase, we ascertain the misclassification frequency, denoted by f, for each image, j, in X^i_{Test}. This frequency is calculated by tallying the occurrences of incorrect classifications for each image across n iterations, expressed as (3).

$$f_j = \frac{\sum^n i = 0(1 \text{ if } y^i True(j) \neq y^i_{pred}(j), \text{ else } 0)}{n} \tag{3}$$



ID-image	$\mathbf{Pred}_{i=0}$	$\cdots$	$\mathbf{Pred}_{i=n}$	f
image-1	0	$\cdots$	1	90%
image-2	1	$\cdots$	0	15%
image-3	1	$\cdots$	1	0%
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

f	f>t
90%	True
15%	False
0%	False
$\vdots$	$\vdots$

Figure 1. Core steps for the ORSAC method

3.3. Flagging and review process for dataset refinement

In this stage of our approach, we concentrate on the flagging and review process integral to dataset refinement. Once the misclassification frequency, f_j, for each image in the dataset is determined through our iterative approach, we employ a threshold, t, to flag images that potentially pose issues. This is formally articulated as (4):

$$for\ j \text{ in } X :\ if\ f_j \geq t,\ then\ Flag_j = True,\ else\ Flag_j = False \tag{4}$$

where $Flag$ signifies the status of an image j being flagged for review.

Images flagged as true are earmarked for further examination or exclusion due to their high frequency of misclassification. The subsequent step involves a manual review of these flagged images to decide whether they should be eliminated or reclassified within the dataset. This manual scrutiny is pivotal to preserve the dataset's integrity as it evolves. Additionally, analyzing these flagged images can offer insights into the reasons behind their frequent misclassification, potentially guiding improvements in the model or data collection methods. In this research, we concentrate solely on flagging and removal rather than flagging and review for a more structured evaluation. Post removal of the flagged images, the refined dataset is denoted as $X^{'}$. Algorithm 1 outlines the pseudocode summarizing the primary steps of our ORSAC methodology. The implementation of our method can be found at: https://github.com/vectech-dev/ORSAC_data_cleaning_public.

Algorithm 1: ORSAC method

```
Input: Dataset $X$
Output: Flagged samples
Divide dataset $X$ into training $X^i_{Train}$, validation $X^i_{Valid}$, and testing $X^i_{Test}$ subsets;
Set $i = 0$; while $i \leq n$ do
    Train a CNN model using $X^i_{Train}$ and evaluate it with $X^i_{Valid}$; Assess the trained CNN on $X^i_{Test}$; Update the
      misclassification frequency, $f$, for each image; Reallocate data splits $(X^i_{Train}, X^i_{Valid}, X^i_{Test})$ using a sliding window
      method; Increment $i$;
end
foreach image in dataset do
    if $f > t$ then
        Flag the image;
    end
end
Flagged images may either be removed or subjected to expert review;
```

4. METHOD

4.1. Method evaluation

To effectively evaluate our methodology, we initially partition the entire raw dataset, X, into two distinct subsets: X_{Final_Test} and X_x. Here, X_{Final_Test} represents a segregated portion of X, reserved exclusively for the final assessment of the model. X_x constitutes the portion of the dataset upon which our method is applied. Within X_x, we generate a modified dataset, $X_{x,ml(h)}$, by intentionally mislabeling a specified percentage, h, of the samples from each class to a different class. Importantly, we retain a record of the original true label for each mislabeled sample. The aforementioned procedures are executed on $X_{x,ml(h)}$ to gauge the effectiveness of the method in identifying mislabeled data.

Further evaluation includes training a model using three distinct dataset variations, subsequently assessed on X_{Final_Test}:

- X_x, the unaltered subset of the dataset designated for method evaluation.
- $X_{x,ml(h)}$, which is X_x but with a fraction of its data, quantified by h, mislabeled.
- $X_{x'}$, the outcome produced by the aforementioned method.

We compare the accuracy of models trained on these three dataset versions using X_{Final_Test}. This comparison not only provides an insight into the method's efficiency but also evaluates its utility in eliminating potentially mislabeled data in scenarios lacking expert review.

The efficacy of our method is quantitatively assessed using key metrics: Recall, defined as the proportion of mislabeled images correctly flagged by our system, and Precision, the proportion of flagged images that were indeed mislabeled. The outcomes are categorized into four types: True positives (mislabeled samples accurately flagged), true negatives (correctly labeled samples not flagged), false positives (correctly labeled but erroneously flagged samples), and false negatives (mislabeled samples not flagged). Additionally, the percentage of the dataset, X_x, that was flagged is reported.

To ascertain the statistical significance of accuracy differences between models trained on $X_{x,ml(h)}$ and $X_{x'}$, a paired difference two-tailed T-test is utilized, calculated as (5):

$$t_{accuracy} = \frac{\overline{D}_{accuracy}}{s_{D_{accuracy}} \big/ \sqrt{n_{accuracy}}} \tag{5}$$

where $\overline{D}_{accuracy}$ is the mean difference in accuracies, $s_{D_{accuracy}}$ is the standard deviation of these differences, and $n_{accuracy}$ is the number of comparisons, all evaluated with an alpha threshold of 0.05.

Furthermore, to analyze significant changes in Precision and Recall when the mislabel frequency is increased from 2.5% to 10% , paired difference directional T-tests are conducted, represented by [43].

$$t_{metric} = \frac{\overline{D}_{metric}}{s_{D_{metric}}/\sqrt{n_{metric}}} \tag{6}$$

With $\overline{D}_{metric}$ being the mean difference in the respective metric,$s_{D_{metric}}$ the standard deviation, and n_{metric} the number of metric comparisons, using an alpha value of 0.05.

4.2. Experimental setup

4.2.1. Datasets

We used three different datasets to evaluate our method: CIFAR-10, which comprises 10 distinct image classes; CIFAR-100, which comprises 100 distinct image classes [44]; and a mosquito dataset from JHU [45], which contains 20 distinct specie. Both CIFAR-10 and CIFAR-100 are composed of 60000 32 × 32 images resized to 224 × 224 using bicubic interpolation , with 50000 in the X_x set , and 10000 in the X_t set. The JHU Mosquito dataset is composed of 10776 images of 480 × 640 pixels, with 9382 in the X_x set and 1394 in the X_t set, and the images were resized to 299 × 299 using bicubic interpolation for training. The minimum number of image samples per species in the mosquito dataset is 142. For all datasets, we conducted experiments with 1% , 2.5% and 10% mislabeled images to test ORSAC under varying levels of label noise. These datasets were selected to showcase ORSAC's adaptability across standard and specialized domains, and to provide a comprehensive evaluation of its performance in different scenarios of label noise.

4.2.2. Network and training details

Each data experiment was run for 35 separate training iterations. A misclassification threshold of 100% was used. This 100% (or concensus) threshold was selected so as to minimize the number of false positives produced by ORSAC. The subsequent evaluations of the models trained on the filtered data used the same configurations listed above, with the exception of the number of epochs and early stopping patience. These were increased to 100 and 20 respectively. For robustness each experiment was repeated 3 times, each time with a unique subset of mislabeled images, though X_{Final_Test} was kept static. The initial order of the dataset was shuffled at the start of each experiment. The averages and standard deviations of the metrics for these experiments are shown in Tables 1 and 2.

For the CIFAR-10 and CIFAR-100 experiments we used an EfficientNet$_{B0}$ [46] model architecture pre-trained on ImageNet. EfficientNet$_{B0}$ was chosen as a lightweight model to align with the low resolution of the CIFAR datasets. In each iteration of training during our cleaning process the batch size was set to 100, and the model was trained using a ranger optimizer [47] with an initial learning rate of 2×10^{-4} , momentum of 0.9, and eps of 1×10^{-6} for 60 epochs with an early stopping patience of 15. A CrossEntropy loss function was used. For the JHU Mosquito dataset experiments we used an Xception model architecture [48] pretrained on ImageNet. Xception was chosen because its depthwise convolutions are known to perform well on high resolution fine-grained datasets such as this one [5], [49], [50]. In each iteration of training the batch size was set at 100, and the model was trained using a ranger optimizer with a learning rate of 5×10^{-4}, momentum of 0.9, and eps of 1×10^{-6} for 60 epochs with an early stopping patience of 15. A focal loss [51] function with a $gamma$ value of 2 was used.

These models were trained using a NVIDIA Corporation RTX A6000 GPU with 48GB of VRAM available. For the JHU Mosquito dataset, with a batch size of 100 using 27GB of VRAM, with the Xception architecture, the approximate time per epoch was 39.7 seconds, resulting in a single training iteration time of 40.7 minutes, and a total training time for a single run of the ORSAC process being 23.7 hours. For the CIFAR-10 dataset, with a batch size of 100 using 18GB of VRAM, with the EfficientNet$_{B0}$ architecture, the approximate time per epoch was 1.6 minutes, resulting in a single training iteration time of 1.7 hours, and a total training time for a single run of the ORSAC process being 58.5 hours. For the CIFAR-10 dataset, with a batch size of 100 using 18GB of VRAM, with the EfficientNet$_{B0}$ architecture, the approximate time per epoch was 1.7 minutes, resulting in a single training iteration time of 1.8 hours, and a total training time for a single run of the ORSAC process being 61.5 hours.

Table 1. Metrics for ORSAC methodology evaluation. h=1%

Dataset	Metric	Value
JHU mosquitoes	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	94.5±.6%
	Recall	95.9±2%
	Precision	30.5±0.2%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	92.9±0.7:92.4±0.4%
	% of $X_{x,ml(h)}$ flagged	3.04±0.09%
CIFAR-10	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	95.0±.1%
	Recall	96.9±.6%
	Precision	30±4%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	93.9±0.2:94.3±0.3%
	% of $X_{x,ml(h)}$ flagged	3.2±0.4%
CIFAR-100	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	79.8±.4%
	Recall	99±1%
	Precision	2.6±.7%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	77.9±0.3:76.8±0.1%
	% of $X_{x,ml(h)}$ flagged	12±1%

Table 2. Metrics for ORSAC methodology evaluation. h=2.5%

Dataset	Metric	Value
JHU mosquitoes	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	94.5±.6%
	Recall	97±1%
	Precision	54±2%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	85.3±0.8: 87.3±0.6%
	% of $X_{x,ml(h)}$ flagged	5.1±0.3%
CIFAR-10	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	95.0±.1%
	Recall	93±1%
	Precision	57±1%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	94.1±0.6: 94.0±0.4%
	% of $X_{x,ml(h)}$ flagged	3.91±0.08%
CIFAR-100	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	79.8±.4%
	Recall	98.4±0.2%
	Precision	20±1%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	78.7±0.4: 78.2±0.4%
	% of $X_{x,ml(h)}$ flagged	12.0±0.3%

5. RESULTS AND DISCUSSION

5.1. Results

The results of the JHU Mosquito, CIFAR-10, and CIFAR-100 dataset with mislabel frequencies, h, of 2.5% and 10% are in Tables 2 and 3 . Respectively for the two h values they achieved Recall of: $97 \pm 1\%$ and $95.08 \pm 0.05\%$; $93 \pm 1\%$ and $92 \pm 1\%$; and $98.4 \pm 0.2\%$ and $98.0 \pm 0.2\%$. The Recall of the ORSAC method in experiments with a 10% mislabel frequency is slightly lower than the Recall of ORSAC in experiments with a 2.5% mislabel frequency for each dataset. However, none of these decreases in Recall are shown to be statistically significant (see Table 4). Respectively for the $h = 2.5\%$ and 10% the datasets (JHU Mosquito, CIFAR-10, and CIFAR-100) achieved Precision of: $54 \pm 2\%$ and $83.2 \pm 0.2\%$; $57 \pm 1\%$ and $81 \pm 3\%$; and $20 \pm 1\%$ and $40.4 \pm 0.6\%$. For every dataset the Precision of the ORSAC method in experiments with $h = 10\%$ is significantly higher than the Precision of ORSAC in experiments with $h = 2.5\%$ (see Table 4). Respectively for the two h values they flagged $5.1 \pm 0.3\%$ and $12.5 \pm 0.3\%$, $3.91 \pm 0.08\%$ and $11.7 \pm 0.5\%$, $12.0 \pm 0.3\%$ and $19.1 \pm 0.1\%$ of the dataset. The accuracy of the model trained on the dataset before and after flagged sample removal (that is, training model M on $X_{x,ml(h)}$ and $X_{x'}$ and testing on X_{Final_Test}) for $h = 2.5\%$ were respectively: 85.3 ± 0.8 and $87.3 \pm 0.6\%$; $94.1 \pm 0.6\%$ and $94.0 \pm 0.4\%$; and 78.7 ± 0.4 and $78.2 \pm 0.4\%$. For the experiment using $h = 10\%$, the respective accuracies were: 83 ± 2 and $86 \pm 2\%$; $93.3 \pm 0.2\%$ and $94.6 \pm 0.3\%$; $77.30 \pm 0.03\%$ and $76.0 \pm 0.2\%$. The baseline accuracy when training model M on the unmodified X_x and testing on X_{Final_Test} for each dataset were respectively: $94.5 \pm 0.6\%$, $95.0 \pm 0.1\%$, $79.8 \pm 0.4\%$. These metrics, are displayed in Tables 2 and 3 with statistical significance results reported in Table 4, while the metrics and statistical significance results for experiments with an h value of 1% are displayed in Table 1 and Table 5 respectively.

The removal of samples flagged by ORSAC resulted in higher accuracy in 3 of the 6 experiments, namely the JHU mosquitoes experiments and the CIFAR-10 experiment with a mislabel frequency of 10%. The CIFAR-100 experiments and the CIFAR-10 experiment with a mislabel frequency of 2.5% all show a slight decrease in accuracy. Of these experiments only the CIFAR-10 and CIFAR-100 experiments with a mislabel frequency of 10% each achieved statistical significance (see Table 4). The difference between the model trained on the unmodified dataset X_x and the unclean dataset $X_{x,ml(h)}$ when testing on the final test set X_{Final_Test} was statistically significantly higher in X_x in all cases except for CIFAR-10 with an $h = 2.5\%$. The t and p values are both measures of the significance of the difference between two datasets. The t value is a measure of the difference between the means of these sets, factoring in the standard deviation, while the p value is the probability of observing that difference assuming that the sets are from the same distribution. A low enough p value indicates that the two datasets are in fact not from the same distribution, which allows us to say that their difference is statistically significant.

Table 3. Metrics for ORSAC methodology evaluation. h=10%

Dataset	Metric	Value
JHU mosquitoes	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	94.5±.6%
	Recall	95.08±0.05%
	Precision	83.2±0.2%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	83±2: 86±2%
	% of $X_{x,ml(h)}$ flagged	12.5±0.3%
CIFAR-10	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	95.0±.1%
	Recall	92±1%
	Precision	81±3%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	93.3±0.2: 94.6±0.3%
	% of $X_{x,ml(h)}$ flagged	11.7±0.5%
CIFAR-100	M^* trained on X_x, tested on $X_{\text{Final\_Test}}$	79.8±.4%
	Recall	98.0±0.2%
	Precision	40.5±0.6%
	M^* trained on $X_{x,ml(h)}$: $X_{x'}$	77.30±0.03: 76.0±0.2%
	% of $X_{x,ml(h)}$ flagged	19.1±0.1%

Table 4. Comparison of metrics using a two-tailed T-test for paired samples

Dataset	Metric	Value (h=2.5%)	Value (h=10%)
JHU mosquitoes	M^* trained on Unmodified (X_x) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-18.06, p=0.003	t=-6.02, p=0.027
	M^* trained on Clean ($X_{x'}$) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=2.08, p=0.173	t=1.14, p=0.37
	Mislabel Frequency Precision	t=28.27, p<0.001	t=-2.59, p=0.061
CIFAR-10	M^* trained on Unmodified (X_x) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-2.70, p=0.114	t=-8.98, p=0.012
	M^* trained on Clean ($X_{x'}$) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-0.05, p=0.97	t=12.86, p=0.006
	Mislabel Frequency Precision	t=15.12, p=0.002	t=-0.55, p=0.32
CIFAR-100	M^* trained on Unmodified (X_x) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-12.59, p=0.006	t=-9.33, p=0.011
	M^* trained on Clean ($X_{x'}$) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-1.05, p=0.40	t=-7.61, p=0.017
	Mislabel Frequency Precision	t=45.90, p<0.001	t=-1.51, p=0.135

Table 5. Comparison of metrics using a two-tailed T-test for paired samples

Dataset	Metric	Value (h=1%)
JHU mosquitoes	M^* trained on Unmodified (X_x) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=6.06, p=0.026
	M^* trained on Clean ($X_{x'}$) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-0.67, p=0.57
CIFAR-10	M^* trained on Unmodified (X_x) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=20.1, p=0.002
	M^* trained on Clean ($X_{x'}$) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=1.62, p=0.25
CIFAR-100	M^* trained on Unmodified (X_x) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=6.44, p=0.023
	M^* trained on Clean ($X_{x'}$) vs Unclean ($X_{x,ml(h)}$) tested on X_{Final_Test}	t=-8.77, p=0.013

5.2. Discussion

Our study presents the ORSAC method's ability to efficiently identify mislabeled images in a dataset. Across all experiments, the lowest percentage of mislabeled images that were found by our process (Recall) was $92 \pm 1\%$, with CIFAR-100 at a mislabeling frequency, or h, of 2.5% reaching $98.4 \pm 0.2\%$ recall. Our testing did not indicate that this metric was significantly affected by the frequency of mislabeled data (see Table 3), but further testing may provide more insights into these dynamics. The Precision of the flagging method, however, was affected by the frequency of mislabeled data, increasing as the mislabeled data frequency increased for each of the datasets. This increased Precision resulted in a smaller portion of the dataset being erroneously flagged for review or removal with respect to h. In the case of flagging data for review, this results in fewer correct samples that must be reviewed by a human expert. Across all experiments, at most $19.1 \pm 0.1\%$ of the dataset was flagged for $h = 10\%$, still dramatically reducing the candidate pool for finding mislabels. Default removal of these flagged samples produced mixed results, as can be seen in the results Tables 1, 2, and 3. A small increase in accuracy was observed in 4 of the 9 experiments, though only the CIFAR-10 dataset at $h = 10\%$ proved significant $(p = 0.006)$, with the other 3 showing a small decrease in accuracy, though only the CIFAR-100 dataset at $h = 10\%$ proved significant $(p = 0.017)$ (see Table 3). Given the low Precision, the lack of significant improvement can be attributed to a loss of variability in the dataset when samples that were in fact correctly labeled (false positives) were removed. For this reason, we recommend that the flagged samples output by ORSAC be reviewed by an expert human reviewer prior to removal or reclassification. The impact of this on resources is illustrated in Figure 2, which plots the percentage of incorrect labels fixed against the percentage of data reviewed.

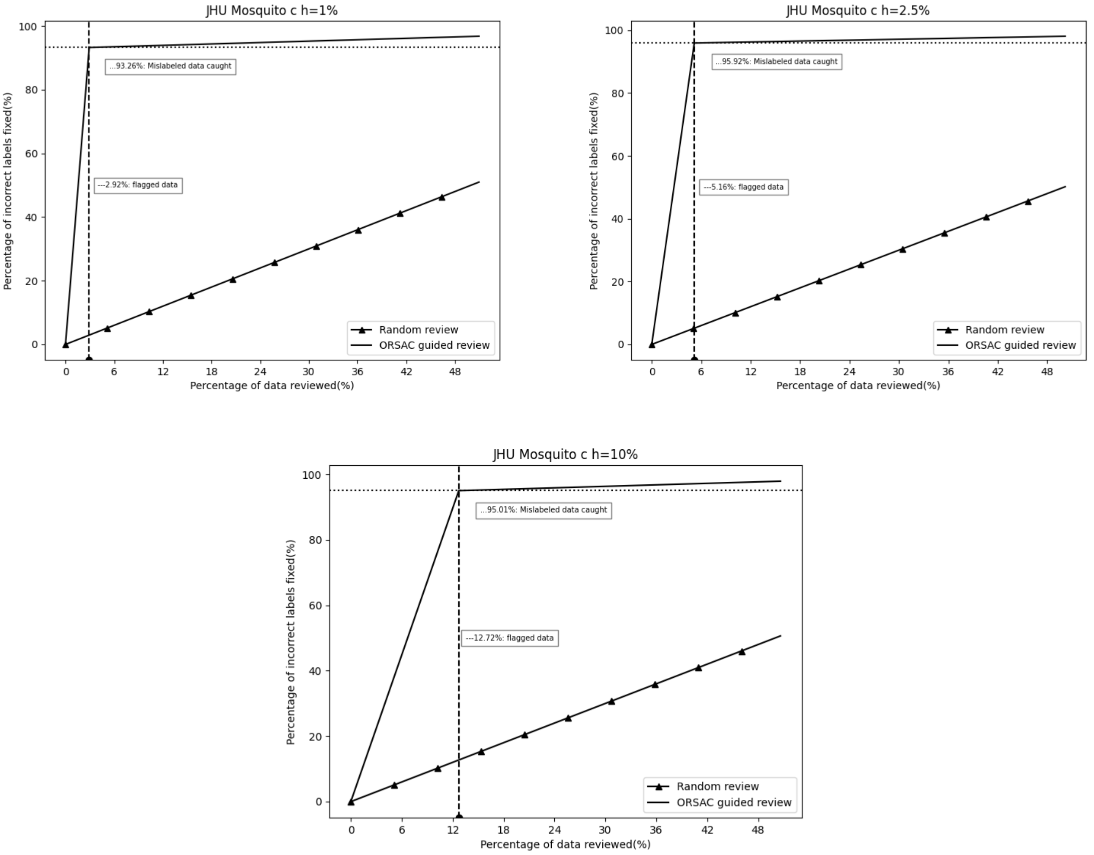


Figure 2. Percentage of incorrect labels fixed plotted against the percentage of data reviewed for experiments performed on the JHU Mosquito dataset with h values of 1%, 2.5%, and 10%, from left to right and top to bottom

While mislabel frequency appears to impact the precision of the method's performance, the dataset also seems to have an influence. CIFAR-10 and the JHU mosquito dataset had very similar Precisions for both mislabel frequencies. The notably reduced Precision observed in CIFAR-100 could be attributed to the heightened complexity of the classification task, stemming from the considerable increase in classes (from 10 in CIFAR-10 to 100 in CIFAR-100), compounded by the notably low resolution of the CIFAR dataset. This low Precision implies that a relatively high accuracy of the model on the modified dataset may be required for optimal functioning of ORSAC. Further research on this point is warranted. The limited impact of increasing label noise on the CIFAR-100 trained model $(X_{x,ml(h)})$ further illustrates this point.

The experiments where $h = 1\%$ illustrate how the importance of clean data persists even in the face of low-label noise. While the effect was small, it was persistent across the datasets. In instances where accuracy is above 90%, an increase of 2% in accuracy has a tangible impact on the perceived error of systems by critical users, amounting to at least a 20% reduction in error frequency. Thus, for systems where users require high accuracy, clean data is paramount. Most publications on data cleaning procedures target high noise environments, with few examining instances of low-label noise [30]. This suggests a research gap of particular importance to computer vision in medicine, entomology, and other fine-grained applications where high accuracy may be of high importance.

Some of the referenced works have very similar methods. The notable SELF method is more computationally frugal than ORSAC, given that it occurs in real-time. However, both SELF and the RANSAC adaptation proposed by Debnath *et al.* [17] use predictions made on data that was in both the train and test set to facilitate their data cleaning processes. This could make it harder to find mislabeled samples, as the model is potentially more likely to label a mislabeled sample as accurate if it has trained on that sample. Our method attempts to circumvent this issue by only using the predictions made by our trained model on the test set. This also limits the variability the ensemble is learning, as the training set remains static through the process. In contrast, ORSAC generates more model variability across models by modifying the training splits each iteration. In addition, relying on predictions from the partially trained model results in lower accuracy of the real-time data cleaning method. This has the potential unwanted effect of diluting the moving prediction average with erroneous predictions.

However, our work isn't without limitations. The success of ORSAC is likely contingent upon the model achieving sufficiently high accuracy on the dataset, which needs further study, though high recall was still achieved in the CIFAR-100 dataset with a baseline accuracy of $79.8 \pm 0.4\%$. In testing ORSAC on CIFAR, a low-resolution coarse-grained classification dataset of 50000 images, and the JHU Mosquito dataset, a high-resolution fine-grained classification dataset of 10776 images, we assess the potential for ORSAC aiding in database development and data cleaning for a wide variety of problems. Notably, the reduction in accuracy from the baseline to the mislabeled data for JHU Mosquitoes, as opposed to the minor reduction from CIFAR-10 and -100, further validates the importance of this issue for fine-grained classification tasks such as taxonomy. However, more testing is required to understand its value for other fine-grained applications and to assess its capability in dealing with very high and very low levels of mislabeling noise (below 1% and above 10%). The computational cost of ORSAC could also be a limiting factor when working with larger datasets or with fewer computational resources. For this case, it may be advantageous for future research to explore the effect of training with lower complexity models and for fewer epochs with regard to speed and accuracy.

6. CONCLUSION

In summation, we presented and tested the ORSAC, for identifying probable mislabels in a dataset. This method, inspired by the well-known RANSAC algorithm, is similar to other methodologies for mislabel detection. Our method is unique in its methodological simplicity, high Recall in finding mislabels, and its capability on disparate computer vision datasets. We also confirm the particular importance of these methods for fine-grained image datasets, applications where high accuracy is essential, and where label noise may exist at low levels. Though only tested in this work on computer vision, the method may be viable for other machine learning classification tasks as well. Future research directions include broadening the application scope of ORSAC to diverse datasets, enhancing its computational efficiency, and increasing its precision without sacrificing its high sensitivity. We advocate for the use of ORSAC primarily as a tool for flagging questionable data, facilitating further review by experts, rather than for outright data exclusion, ensuring a balanced approach to maintaining data integrity.

ACKNOWLEDGEMENTS

The authors would like to sincerely thank other members of the Vectech team who each provided support in the development of these methods on the Vectech IDX dataset, in particular Roy Faiman, Sanket Padmanabhan, and Rongjun Wu. Furthermore, we would like to thank all the contributors to the IDX dataset, without whom we would not have prototyped this method. This work was funded by the National Science Foundation Small Business Innovation Research (SBIR) grants, FAIN 2039534 (Phase I), and 2322335 (Phase II), and the National Institutes of Health SBIR grant, 2R44AI162425-02 (Phase II).

REFERENCES

[1] X. S. Wei *et al.*, "Fine-grained image analysis with deep learning: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8927–8948, 2022, doi: 10.1109/TPAMI.2021.3126648.

[2] L. Zhang, S. Huang, W. Liu, and D. Tao, "Learning a mixture of granularity-specific experts for fine-grained categorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, vol. 2019-October, pp. 8330–8339, doi: 10.1109/ICCV.2019.00842.

[3] Q. X. Huang, H. Su, and L. Guibas, "Fine-grained semi-supervised labeling of large shape collections," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1–10, 2013, doi: 10.1145/2508363.2508364.

[4] C. O. Coleman, "Taxonomy in times of the taxonomic impediment - examples from the community of experts on amphipod crustaceans," *Journal of Crustacean Biology*, vol. 35, no. 6, pp. 729–740, 2015, doi: 10.1163/1937240X-00002381.

[5] A. Goodwin *et al.*, "Mosquito species identification using convolutional neural networks with a multitiered ensemble model for novel species detection," *Scientific Reports*, vol. 11, no. 1, p. 13656, 2021, doi: 10.1038/s41598-021-92891-9.

[6] C. A. of S. USA, "AntWeb," *CABI Compendium-Compendium Identification Guides*, no. Identification Guides, p. 20187200669, 2018, doi: 10.5555/cabicompendium.20187200669.

[7] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, "Birdsnap: large-scale fine-grained visual categorization of birds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2019–2026, doi: 10.1109/CVPR.2014.259.

[8] R. R. Klopper, G. F. Smith, and A. C. Chikuni, "The global taxonomy initiative in Africa," Taxon, vol. 51, no. 1, pp. 159–165, 2002, doi: 10.2307/1554974.

[9] G. H. Dar, A. A. Khuroo, C. S. Reddy, and A. H. Malik, "Impediment to taxonomy and its impact on biodiversity science: an Indian perspective," *Proceedings of the National Academy of Sciences India Section B - Biological Sciences*, vol. 82, no. 2, pp. 235–240, 2012, doi: 10.1007/s40011-012-0031-3.

[10] J. La Salle, Q. Wheeler, P. Jackway, S. Winterton, D. Hobern, and D. Lovell, "Accelerating taxonomic discovery through automated character extraction," *Zootaxa*, vol. 2217, no. 2217, pp. 43–55, 2009, doi: 10.11646/zootaxa.2217.1.3.

[11] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: exploring techniques and remedies in medical image analysis," *Medical Image Analysis*, vol. 65, p. 101759, 2020, doi: 10.1016/j.media.2020.101759.

[12] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning–based data anomaly detection method for structural health monitoring," *Structural Health Monitoring*, vol. 18, no. 2, pp. 401–421, 2019, doi: 10.1177/1475921718757405.

[13] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365, 2015*, doi: 10.48550/arXiv.1506.03365.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981, doi: 10.1145/358669.358692.

[15] E. Brachmann and C. Rother, "Neural-guided RANSAC: learning where to sample model hypotheses," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, vol. 2019-October, pp. 4321–4330, doi: 10.1109/ICCV.2019.00442.

[16] C.-H. Tsai and Y.-S. Peng, "Unsupervised image outlier detection using RANSAC," *arXiv preprint arXiv:2307.12301*, 2023, doi: 10.48550/arXiv.2307.12301.

[17] S. Debnath, A. Banerjee, and V. Namboodiri, "Adapting RANSAC SVM to detect outliers for robust classification," in *BMVC*, 2015, pp. 168.1-168.11, doi: 10.5244/c.29.168.

[18] J. Zhang, N. Inkawhich, R. Linderman, Y. Chen, and H. Li, "Mixture outlier exposure: towards out-of-distribution detection in fine-grained environments," in *Proceedings - 2023 IEEE Winter Conference on Applications of Computer Vision, WACV 2023*, 2023, pp. 5520–5529, doi: 10.1109/WACV56688.2023.00549.

[19] P. Trivedi, T. Agarwal, and K. Muthunagai, "MC-RANSAC: a pre-processing model for RANSAC using Monte Carlo method implemented on a GPU," in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, 2013, pp. 1380–1383, doi: 10.1109/ICACCI.2013.6637380.

[20] C. S. Chen and Y. P. Hung, "RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–1234, 1999, doi: 10.1109/34.809117.

[21] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm, "USAC: a universal framework for random sample consensus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 2022–2038, 2013, doi: 10.1109/TPAMI.2012.257.

[22] B. Soni, P. K. Das, and D. M. Thounaojam, "CMFD: a detailed review of block based and key feature based techniques in image copymove forgery detection," *IET Image Processing*, vol. 12, no. 2, pp. 167–178, 2018, doi: 10.1049/iet-ipr.2017.0441.

[23] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6492 LNCS, no. PART 1, pp. 135–148, doi: 10.1007/978-3-642-19315-6_11.

[24] R. Cupec, R. Grbić, E. K. Nyarko, K. Sabo, and R. Scitkovski, "Detection of planar surfaces based on RANSAC and LAD plane fitting," in *Proceedings of the 4th European Conference on Mobile Robots - ECMR'09*, 2009, pp. 37–42.

[25] M. Fan, S. W. Jung, and S. J. Ko, "Highly accurate scale estimation from multiple keyframes using RANSAC plane fitting with a novel scoring method," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15335–15345, 2020, doi: 10.1109/TVT.2020.3040014.

[26] Harintaka and C. Wijaya, "Automatic point cloud segmentation using RANSAC and DBSCAN algorithm for indoor model," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 21, no. 6, pp. 1317–1325, Dec. 2023, doi: 10.12928/TELKOMNIKA.V21I6.25299.

[27] A. Bucci, L. Zacchini, M. Franchi, A. Ridolfi, and B. Allotta, "Comparison of feature detection and outlier removal strategies in a mono visual odometry algorithm for underwater navigation," *Applied Ocean Research*, vol. 118, p. 102961, 2022, doi: 10.1016/j.apor.2021.102961.

[28] J. Zheng, W. Peng, Y. Wang, and B. Zhai, "Accelerated RANSAC for accurate image registration in aerial video surveillance," *IEEE Access*, vol. 9, pp. 36775–36790, 2021, doi: 10.1109/ACCESS.2021.3061818.

[29] W. Guilluy, L. Oudre, and A. Beghdadi, "Video stabilization: overview, challenges and perspectives," *Signal Processing: Image Communication*, vol. 90, p. 116015, 2021, doi: 0.1016/j.image.2020.116015.

[30] K. G. Moura, R. B. C. Prudêncio, and G. D. C. Cavalcanti, "Label noise detection under the noise at random model with ensemble filters," *Intelligent Data Analysis*, vol. 26, no. 5, pp. 1119–1138, 2022, doi: 10.3233/IDA-215980.

[31] W. Feng, Y. Quan, and G. Dauphin, "Label noise cleaning with an adaptive ensemble method based on noise detection metric," *Sensors (Switzerland)*, vol. 20, no. 23, pp. 1–16, 2020, doi: 10.3390/s20236718.

[32] P. Wu, S. Zheng, M. Goswami, D. Metaxas, and C. Chen, "A topological filter for learning with label noise," *Advances in Neural Information Processing Systems*, vol. 2020-December, pp. 21382–21393, 2020, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/f4e3ce3e7b581ff32e40968298ba013d-Paper.pdf.

[33] A. Ghorbani and J. Zou, "Data shapley: equitable valuation of data for machine learning," in *36th International Conference on Machine Learning*, ICML 2019, 2019, vol. 2019-June, pp. 4053–4065, [Online]. Available: https://proceedings.mlr.press/v97/ghorbani19c/ghorbani19c.pdf.

[34] Y. J. Koh, C. Lee, and C. S. Kim, "Video stabilization based on feature trajectory augmentation and selection and robust mesh grid warping," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5260–5273, 2015, doi: 10.1109/TIP.2015.2479918.

[35] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. Phuong Nguyen, L. Beggel, and T. Brox, "Self: learning to filter noisy labels with self-ensembling," *8th International Conference on Learning Representations, ICLR 2020*, 2020, doi: 10.48550/arXiv.1910.01842.

[36] R. Zhou, W. Gan, F. Wang, Z. Yang, Z. Huang, and H. Gan, "Tri-correcting: label noise correction via triple CNN ensemble for carotid plaque ultrasound image classification," *Biomedical Signal Processing and Control*, vol. 91, p. 105981, 2024, doi: 10.1016/j.bspc.2024.105981.

[37] D. Park, S. Choi, D. Kim, H. Song, and J. G. Lee, "Robust data pruning under label noise via maximizing re-labeling accuracy," in *Advances in Neural Information Processing Systems*, 2023, vol. 36, pp. 74501–74514, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/ebb6bee50913ba7e1efeb91a1d47a002-Paper-Conference.pdf.

[38] M. T. Aziz, S. M. H. Mahmud, K. O. M. Goh, and D. Nandi, "Addressing label noise in leukemia image classification using small loss approach and pLOF with weighted-average ensemble," *Egyptian Informatics Journal*, vol. 26, p. 100479, 2024, doi: 10.1016/j.eij.2024.100479.

[39] L. Zeng, X. Chen, X. Shi, and H. T. Shen, "Feature noise boosts DNN generalization under label noise," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2024, doi: 10.1109/TNNLS.2024.3394511.

[40] R. Biswas, P. Malmsköld, and L. Davidson, "GPU-accelerated computational methods using Python and CUDA," 2024, [Online]. Available: https://weilong-web.github.io/data/CUDA for CFD.pdf.

[41] R. Hunter, S. Ross, and J.-R. Cheng, "A general-purpose multiplatform GPU-accelerated ray tracing API," Jul. 2023. doi: 10.21079/11681/47260.

[42] W. S. Moses, I. R. Ivanov, J. Domke, T. Endo, J. Doerfert, and O. Zinenko, "High-performance GPU-to-CPU transpilation and optimization via high-level parallel constructs," *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP*, pp. 119–134, 2023, doi: 10.1145/3572848.3577475.

[43] M. C. Gonçalves and R. Silva, "The effect of statistical hypothesis testing on machine learning model selection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2023, vol. 14196 LNAI, pp. 415–427, doi: 10.1007/978-3-031-45389-2_28.

[44] A. Krizhevsky and G. Hinton, "CIFAR-10/100 (Canadian Institute for Advanced Research)." 2009, URL http://www. cs. toronto. edu/kriz/cifar. html.

[45] A. Goodwin *et al.*, "Mosquito species identification using convolutional neural networks with a multitiered ensemble model for novel species detection," *Scientific Reports*, vol. 11, no. 1. 2021, doi: 10.1038/s41598-021-92891-9.

[46] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[47] L. Wright and N. Demeure, "Ranger21: a synergistic deep learning optimizer," *arXiv preprint arXiv:2106.13731*, 2021, doi: 10.48550/arXiv.2106.13731.

[48] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January. IEEE, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[49] Z. Zhao, Z. Luo, J. Li, K. Wang, and B. Shi, "Large-scale fine-grained bird recognition based on a triplet network and bilinear model," *Applied Sciences (Switzerland)*, vol. 8, no. 10, p. 1906, 2018, doi: 10.3390/app8101906.

[50] Q. Zhang, J. Sang, W. Wu, Z. Wu, H. Xiang, and B. Cai, "Fine-grained image classification based on Xception," *Chongqing Daxue Xuebao/Journal of Chongqing University*, vol. 41, no. 5, pp. 85–91, 2018, doi: 10.11835/j.issn.1000-582X.2018.05.011.

[51] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, vol. 42, no. 2, pp. 318–327, doi: 10.1109/TPAMI.2018.2858826.

BIOGRAPHIES OF AUTHORS

Thomas Jenkins received his bachelor's degree in computer science and physics from Brown University in Providence Rhode Island in May 2021. He is currently a computer vision engineer at Vectech. His primary areas of interest are artificial intelligence, computer vision, deep learning, and data science. He can be contacted at email: thomas@vectech.io.

Autumn Goodwin is a Co-founder and the Chief Technology Officer of Vectech. Autumn Goodwin earned an MSE from the Center of Bioengineering Innovation and Design at Johns Hopkins University. Autumn Goodwin have since focused their research and development efforts on making new vector surveillance tools to support public and environmental health. Autumn Goodwin work engages the whole lifecycle of development of practical computer vision tools, including stakeholder needs assessment, imaging design, database development, model development, model deployment, and the refinement of datasets and models for robust and stable deployments. Autumn Goodwin are interested in designing and developing solutions to public and environmental health problems, in particular as they relate to small arthropods and AI. Autumn Goodwin can be contacted at email: autumn@vectech.io.

Sameerah Talafha a Computer Vision Engineer at Vectech, excels in generative AI, deep learning, and pattern recognition. Sameerah earned a Ph.D. from Southern Illinois University in 2022. Sameerah's interest lies in advancing AI technologies, particularly in the realm of computer vision. At Vectech, Sameerah focuses on innovating and refining AI models, demonstrating a commitment to exploring the potential of AI applications. He can be contacted at email: sameerah.talafha@siu.edu.