technologies

MDPI

Article

Identifying and Mitigating Label Noise in Deep Learning for Image Classification

César González-Santoyo [1], Diego Renza [1,*] and Ernesto Moya-Albor [2]

[1] Facultad de Ingeniería, Universidad Militar Nueva Granada, Carrera 11 101-80, Bogota 110111, DC, Colombia; est.cesare.gonzalez@unimilitar.edu.co
[2] Facultad de Ingeniería, Universidad Panamericana, Augusto Rodin 498, Mexico City 03920, Mexico; emoya@up.edu.mx
* Correspondence: diego.renza@unimilitar.edu.co

Abstract: Labeling errors in datasets are a persistent challenge in machine learning because they introduce noise and bias and reduce the model's generalization. This study proposes a novel methodology for detecting and correcting mislabeled samples in image datasets by using the Cumulative Spectral Gradient (CSG) metric to assess the intrinsic complexity of the data. This methodology is applied to the noisy CIFAR-10/100 and CIFAR-10n/100n datasets, where mislabeled samples in CIFAR-10n/100n are identified and relabeled using CIFAR-10/100 as a reference. The DenseNet and Xception models pre-trained on ImageNet are fine-tuned to evaluate the impact of label correction on the model performance. Evaluation metrics based on the confusion matrix are used to compare the model performance on the original and noisy datasets and on the label-corrected datasets. The results show that correcting the mislabeled samples significantly improves the accuracy and robustness of the model, highlighting the importance of dataset quality in machine learning.

Keywords: mislabeled samples; spectral clustering; image classification; deep learning; noisy labels

check for updates

Academic Editors: George F. Fragulis and Debopriyo Roy

Received: 26 February 2025
Revised: 21 March 2025
Accepted: 26 March 2025
Published: 1 April 2025

Citation: González-Santoyo, C.; Renza, D.; Moya-Albor, E. Identifying and Mitigating Label Noise in Deep Learning for Image Classification. *Technologies* **2025**, *13*, 132. https://doi.org/10.3390/technologies13040132

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, the progress in artificial intelligence models has been remarkable, driven primarily by the increasing availability of large datasets and improvements in computing power [1]. In particular, labeled datasets have been instrumental in enabling the training of supervised learning algorithms for tasks such as image classification, object detection, or image segmentation. Well-known datasets such as MNIST, CIFAR-10, and ImageNet are widely used in the deep learning community as benchmarks to evaluate and compare the performance of newly developed models [2]. However, the utility of these datasets is closely related to the accuracy of their labels, as labeling errors can significantly impact the model performance.

Recent studies have shown that even low label error rates can significantly affect the model's generalization and accuracy, further highlighting the importance of addressing label noise in datasets [3–6]. As a result, research on label noise detection and correction has become increasingly important to ensure the reliability and robustness of artificial intelligence systems. Labeling errors are a common challenge in various types of data, including text, audio, image, and video. These errors can distort the learning process, introduce noise, and reduce the generalization capabilities of machine learning models. Even widely used datasets such as MNIST, CIFAR-10, and ImageNet are not exempt from

these problems, highlighting the need for effective methods to detect and correct mislabeled samples [3,7].

Detecting and correcting mislabeled samples are critical steps in data preprocessing, as these anomalies can introduce bias and noise into the learning process. Techniques such as supervised learning and anomaly detection algorithms have been developed to systematically identify and correct these errors [3]. By ensuring data quality through accurate labeling and a deeper understanding of the class relationships, more robust and efficient convolutional neural networks (CNNs) can be developed. Addressing this issue not can only improve the model accuracy but also can reduce the computational complexity by reducing the need to retrain models with noisy data [4,5].

The impact of label noise on classification problems has been addressed in the literature for several areas of classification. One study recognized the two types of label noise in medical images, disagreement and single-target label noise, and related them to the model performance on dermatology, ophthalmology, and pathology datasets [8]. It is also possible to find research that has addressed the label noise problem for segmentation applications in clinical settings [9]. A recent study that provides an overview of label noise learning for medical image analysis can be found in [10]. In remote sensing, problems with label noise may include studies based on single-label classification or multi-label scene classification for land cover mapping [11,12] or hyperspectral image classification applications [13].

Several approaches have been proposed to addressing the data quality issues caused by labeling errors or biased labels. Methods based on a constrained optimization framework have been proposed that apply a training sample reweighting method to learn an equivalent unbiased labeling function [14]. Alternative approaches have attempted to identify instances where a previously trained classifier (on the same data) assigns a very low probability to the original label, suggesting that the model considers these instances to be mislabeled and in need of revision [5]. Confidence-based learning removes erroneous samples from the dataset by applying a probabilistic threshold to the confidence scores given by a trained model; with the cleaned dataset, the model is retrained to improve its robustness [7].

A variety of approaches have also been proposed for learning with noisy labels. In DivideMix, semi-supervised learning is used to train two networks simultaneously by coding datasets, co-refinement, and co-division [15]. In CORES (COnfidence REgularized Sample Sieve), corrupted examples are filtered out in order to treat clean and corrupted examples separately when training a deep learning model [16]. On the other hand, methods such as PES (Progressive Early Stopping) [17] or ELR+ (Early-Learning Regularization) [18] propose techniques to prevent deep learning models from memorizing the labels for examples with noisy labels, either by controlling the early stopping point or by applying regularization techniques.

Researchers at MIT conducted a remarkable study, known as Cleanlab, on analyzing ten widely used datasets, including MNIST, CIFAR-10, and ImageNet. Using reinforcement learning, they proposed a method specifically designed to detect and correct mislabeled data, identifying numerous labeling inaccuracies. These results were later validated using Amazon's Mechanical Turk platform to ensure the reliability of the corrections [3,7].

Many of these approaches rely on deep learning model training, which are effective but often require significant computational resources. That is, despite advances in labeling error detection and correction, many of the existing techniques require large computational resources, limiting their applicability in resource-constrained scenarios. This limitation underscores the importance of exploring alternative strategies that are efficient and scalable, allowing for the accurate detection of mislabeled or atypical samples without an excessive computational overhead.

However, characterization of the datasets prior to training machine learning models is critical to ensure the optimal performance. Several studies have focused on quantifying the intrinsic complexity of the data to identify the challenges that models may face during training [19–21]. Analysis of the interclass similarity has proven to be essential, as it highlights potential overlaps and areas of confusion between classes. High interclass similarity can lead to increased misclassification rates, ultimately reducing the overall accuracy of CNNs [3,5,19,21]. By quantifying these similarities, researchers can design more robust learning strategies and refine datasets to improve the generalization of machine learning models.

Therefore, in this paper, we propose a new methodology for detecting errors in the labeling of image datasets based on an evaluation of the intrinsic complexity of the data using the Cumulative Spectral Gradient (CSG) metric. Unlike traditional approaches based on confident learning, the use of the CSG allows for the quantification of the probabilistic divergence between classes within a spectral clustering framework, which facilitates the identification of samples that do not conform to the expected distribution. By identifying and correcting these anomalies, we aim to significantly improve the generalization of deep learning models, especially in scenarios where the data quality is a limiting factor. The main contribution of this work is to identify samples with labeling errors without the need to train a machine learning model. This reduces the time associated with the data analysis phase.

The rest of this article is organized as follows: Section 2 presents an explanation of the basic issues in this article related to the comparison metrics, deep learning architecture, and datasets. Section 3 describes the proposed methodology in detail, including the selected datasets, the scoring system, the relabeling, and the fine-tuning of the DL model. Section 4 presents the experimental results obtained and an analysis of them. Finally, Section 5 presents the conclusions of this study.

2. The Background

An explanation of the metric for dataset complexity used in the proposed method, as well as the variant of the CIFAR-10/100 datasets and the pre-trained models used to validate the proposed method, is presented below.

2.1. The Cumulative Spectral Gradient Metric

In the context of data evaluation, computational efficiency is a critical factor. The CSG metric provides a scalable and efficient solution for quantifying the complexity of multi-class classification problems. By avoiding complex and costly computations, the CSG allows large datasets to be evaluated quickly and accurately. Moreover, its close relationship with the generalization capability of CNNs makes it an invaluable tool for model selection and hyperparameter optimization.

The calculation of the CSG metric is based on the analysis of the eigenvalue spectrum obtained from a similarity matrix between samples. An eigenvalue spectrum with low eigenvalues indicates a clear separation between classes, suggesting that the samples in each class are compactly grouped into the feature space. In contrast, a spectrum with high eigenvalues shows more overlap between classes, making the classification task more difficult [19,22].

The CSG method first projects the input data into a low-dimensional latent space. This nonlinear projection aims to match the most relevant features of the data with the learning capacity of the models. During this transformation, samples belonging to the same class tend to cluster into compact regions of the latent space, while samples from different classes separate. This divergence between the distributions of two classes (C_i, C_j) can be

approximated by averaging the probability that a given number of samples (M) from one class is in the other class (1) [19]:

$$E \approx \frac{1}{M} \sum_{m=1}^{M} P(\phi(x)|C_j), \tag{1}$$

where $\phi(x)$ represents the M samples drawn from the identically and independently distributed class. This model is approximated using a K-nearest estimator according to Equation (2):

$$E \approx \frac{K_{C_j}}{M \cdot V}, \tag{2}$$

where K_{C_j} is the number of neighbors around $\phi(x)$ of class C_j, M is the number of samples selected in class C_j, and V is the volume of the hypercube surrounding the k samples closest to $\phi(x)$ in class C_j [19].

After calculating the divergences between all classes, a square similarity matrix S is obtained, where each element of this matrix represents the approximation of the divergence between the classes C_i and C_j. Then, each column of the matrix S is considered a vector of signatures of each class i, and an adjacency matrix W is computed using the Bray–Curtis distance function (Equation (3)):

$$w_{ij} = 1 - \frac{\sum_k^K |S_{ik} - S_{jk}|}{\sum_k^K |S_{ik} + S_{jk}|}, \tag{3}$$

Each of the weights w_{ij} of the adjacency matrix encodes the degree of connection between the classes (the larger the value, the higher the degree of connection). Using these values, we compute the degree matrix (D) and the Laplacian matrix (L), given by Equations (4) and (5):

$$D_i = \sum_j w_{ij}. \tag{4}$$

$$L = D - W. \tag{5}$$

The set of n eigenvalues of the matrix L (λ_i) is called the spectrum of L, over which the gradient discontinuity (or eigengap) is calculated ($\lambda_{i+1} - \lambda_i$). Finally, the calculation of the total complexity of the dataset takes into account the area under the spectrum curve and the position of the eigengap and corresponds to the cumulative maximum (*cummax*) of the eigengaps normalized by $K - i$ [19,23]:

$$\mathrm{CSG} = \sum_i \mathrm{cummax}\left(\frac{\lambda_{i+1} - \lambda_i}{K - i}\right)_i. \tag{6}$$

2.2. The DenseNet Architecture

DenseNet is a type of CNN architecture that connects each layer to every other layer in a feedforward manner to reduce both the number of parameters and the decreasing gradient problem and to improve the feature propagation and feature reuse [24].

The main units that make up DenseNet are dense blocks and transition layers. The dense blocks define how the inputs and outputs are concatenated. These dense blocks consist of a series of convolutional layers, including batch normalization and the ReLU activation function, where each layer has the same number of channels and also takes as input all of the feature maps provided by the previous layers. The transition layer consists of a batch normalization layer and a 1×1 convolutional layer, followed by a

2 × 2 averaging pooling layer. In addition, this layer controls the number of channels to reduce the complexity of the model.

The DenseNet architecture can be built with different levels of depth. The DenseNet121, DenseNet169, and DenseNet2021 versions are available as pre-trained models on ImageNet with a top-1 accuracy of 75.0%, 76.2%, and 77.3%, respectively. The general structure of the DenseNet architecture is shown in Figure 1 [24,25].

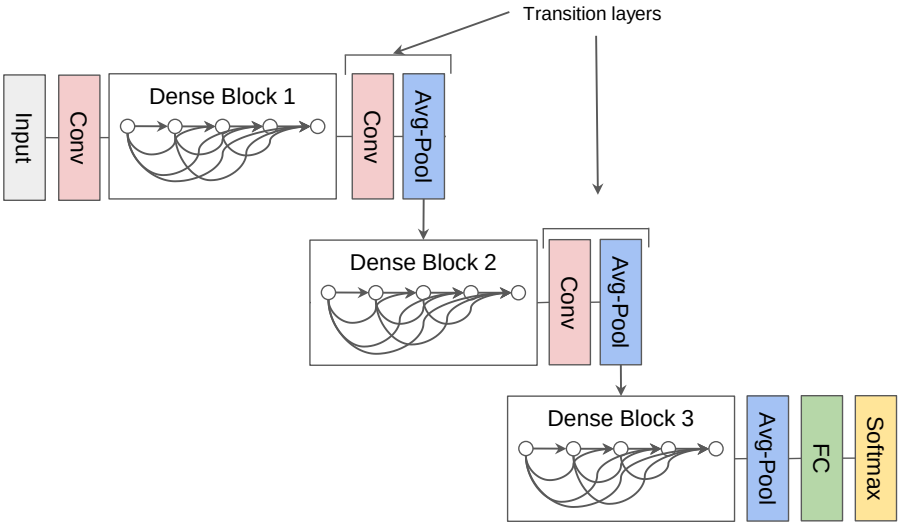


Figure 1. DenseNet architecture.

2.3. Xception Architecture

Xception is a CNN architecture characterized by a linear stack of depthwise separable convolution layers, integrated with residual connections. Each depthwise separable convolution consists of two steps: first, spatial convolution is applied independently to each input channel; then, a 1 × 1 convolution is used to project the resulting channels into a new channel space [26].

The Xception architecture uses 36 convolutional layers as the basis for the network feature extraction. These layers are organized into 14 modules, with linear residual connections surrounding modules 2 through 13. Behind the feature extraction base is a logistic regression layer designed for image classification [26]. The Xception architecture is available as a pre-trained model on ImageNet with a top-1 accuracy of 79.0%.

2.4. Mislabeled Sample Dataset Generation (CIFAR-n)

CIFAR-10n and CIFAR-100n are variants of the well-known CIFAR-10 and CIFAR-100 datasets. These new versions incorporate a level of noise into the labels using human annotations obtained through Amazon Mechanical Turk (Mturk). This process simulates real-world conditions where the labels may be inaccurate or contain errors, making these datasets a valuable tool for evaluating the robustness of machine learning algorithms in the presence of label noise [27].

To generate CIFAR-10n, the CIFAR-10 training set was randomly divided into ten disjoint subsets. Each subset was presented on the Amazon Mechanical Turk (Mturk) platform as a series of Human Intelligence Tasks (HITs), where each HIT contained ten images. Each HIT was assigned to three independent workers who provided a label for each image. Thus, each image in the original training set received three noisy labels provided by the Mturk workers (in addition to the clean image label). From these noisy labels, five sets of noisy labels were constructed, aggregate, random *i* (1, 2, 3), and worst, with each representing different levels and types of noise [27].

To generate the aggregate label set, a majority voting system was applied to the three noisy labels provided by the Mturk workers for each image. In case of a disagreement, one label was randomly selected from the three choices. The random i label sets (i = 1, 2, 3) were created by randomly selecting one of the three noisy labels for each image, ensuring that each worker labeled different images. Finally, the worst label set was constructed by assigning each image the label that most closely matched the clean label, simulating a high-noise scenario. If none of the noisy labels were incorrect, the original clean label was retained [27].

The results of the labeling process in CIFAR-10n showed high agreement between the labelers in about 60% of the images. However, the generated noisy label sets showed a wide range of noise levels. The aggregate set, combining the three labels, had the lowest noise index (9.03%). The random sets showed moderate noise indices (between 17.23% and 18.12%), while the worst set achieved a significantly higher noise value (40.21%), demonstrating the effectiveness of this set for simulating high-noise scenarios [27]. Some example images of CIFAR-10n worst are shown in Figure 2.

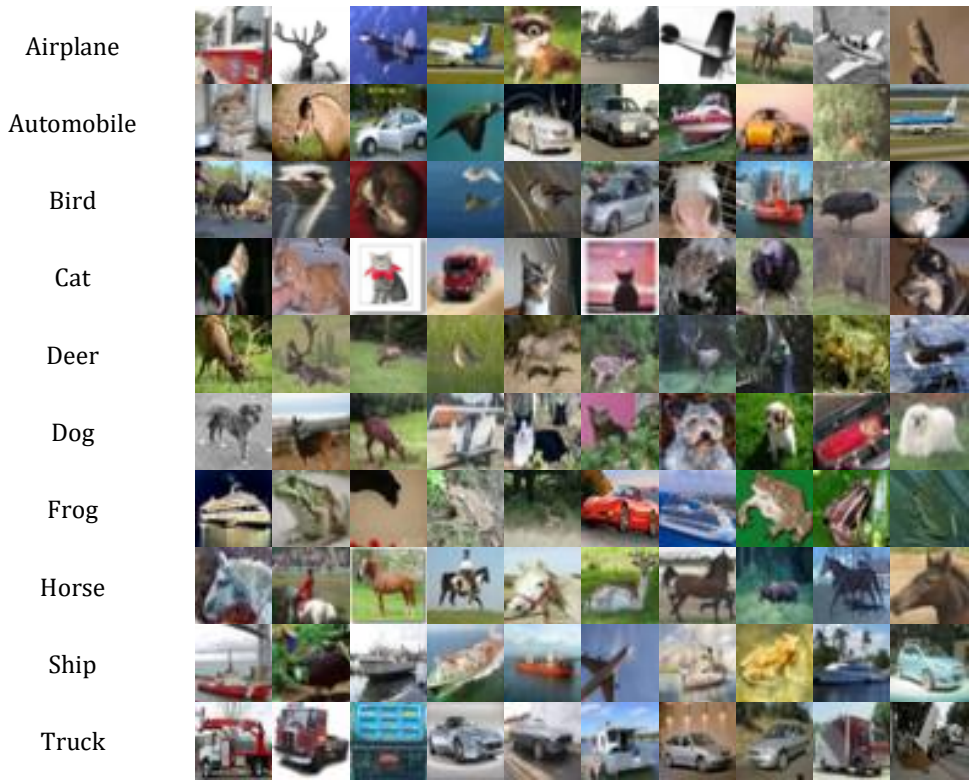


Figure 2. Sample images of the CIFAR-10n worst dataset.

As for CIFAR-100n, the training dataset was divided into ten batches without replacement with five images per HIT. The 100 classes were grouped into 20 disjoint superclasses, similar to the 20 coarse categories given in CIFAR-100. Each HIT had a worker who first selected the superclass for each image and then the fine label, using an example image as a reference [27].

Thus, each image in CIFAR-100n contained a coarse label and a fine label given by a human annotator. Such annotations resulted in a total noise level corresponding to 25.6% for the coarse labels and 40.2% for the fine labels [27].

3. Materials and Methods

The proposed method is based on the application of the CSG metric, which allows a confidence score to be obtained that facilitates relabeling a given number of samples according to their score. In this study, the CIFAR-10/100 and CIFAR-10n/100n datasets (using the worst case) are analyzed and compared class by class to identify mislabeled samples. This means that the complexity of each pair of classes is evaluated by applying the CSG metric (using a subset consisting of class *i* from CIFAR-10/100 and class *i* from CIFAR-10n/100n). When the CSG is applied to each pair of classes, each of the CIFAR-10n/100n samples contributes to a certain degree to the complexity of this subset. This degree of complexity is used as a confidence score to select N samples from each class for relabeling. This process is repeated for all classes, resulting in a new version of the dataset with N corrected samples per class.

Subsequently, a deep learning model is trained on the newly created dataset, and its performance can be compared to the performance of training the same model on the original CIFAR-10/100 and the CIFAR-10n/100n datasets. This comparison allows for an evaluation of the impact of label correction on the model performance, providing insights into the effectiveness of the proposed methodology in improving the dataset quality and consequently the accuracy and robustness of the deep learning model. The proposed methodology is illustrated in Figure 3, with the details provided below.

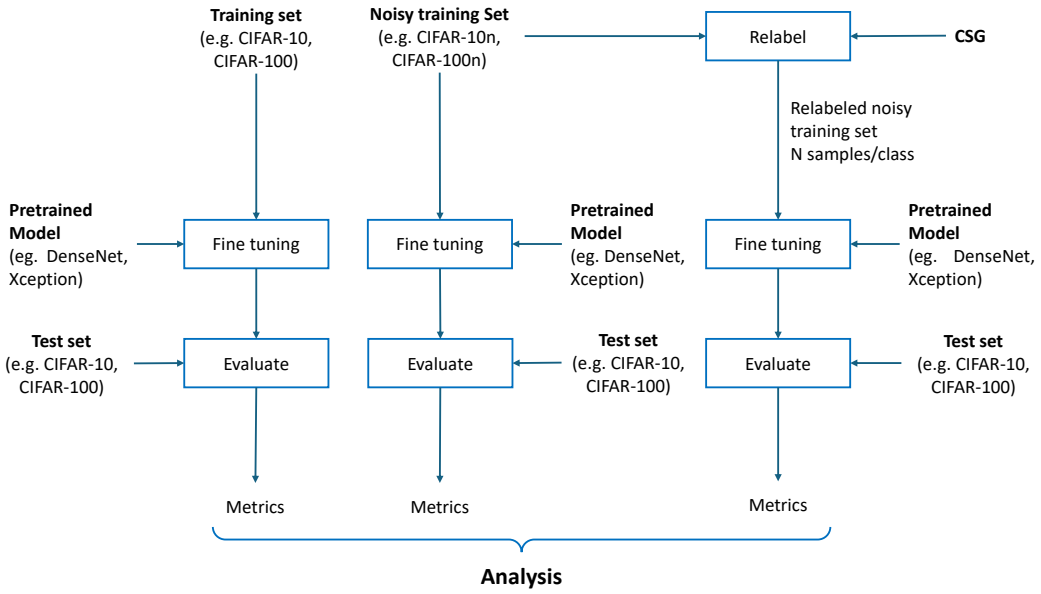


Figure 3. Outline of the proposed methodology.

3.1. Datasets

This research was conducted using the CIFAR-10, CIFAR-100, CIFAR-10n-Worst, and CIFAR-10n-noisy datasets. Since the goal is to identify mislabeled images in the CIFAR-10n-Worst and CIFAR-10n-noisy datasets (with about 40% noise in the labels), the most viable approach is to compare the performance of the models with respect to their performance on the original CIFAR-10/100 data.

It is important to note that the subset of training data in CIFAR-10/100 contains 5000 and 500 samples in each class, respectively, while in CIFAR-10n/100n, the number of training images per class is variable. For example, the car class in CIFAR-10n-worst has the largest number of samples (6053), and the deer class has the fewest (4040). In addition, each of the classes in CIFAR-10n/100n represents a mixture of images that include not only the respective class but also multiple images from other classes as a result of the labeling

process explained above. In any case, the total number of images in the dataset is always the same (50,000 samples). Both CIFAR-10 and CIFAR-100 have an additional test subset of 10,000 images each, i.e., 1000 test images per class in CIFAR-10 and 100 test images per class in CIFAR-100.

Considering that the proposed methodology consisted of a data-centric approach, the training of the models was performed using the entire training set (50,000 images in both CIFAR-10 and CIFAR-100), i.e., no validation subset was used for hyperparameter fitting. The original test set (10,000 images in both CIFAR-10 and CIFAR-100) was used for the model evaluation.

3.2. The Confidence Scores

The CSG metric was applied to subsets consisting of one class from CIFAR-10 (train) and the corresponding class from CIFAR-10n. For example, the aircraft class from CIFAR-10 was compared to the aircraft class from CIFAR-10n-Worst, followed by comparing the automobile class from CIFAR-10 to the automobile class from CIFAR-10n-Worst, and so on for all remaining classes. This approach ensured a direct and meaningful comparison between the datasets and facilitated the identification of mislabeled samples in CIFAR-10n-Worst. A similar process was used for CIFAR-100 and CIFAR-100n but using their 100 classes.

In addition, the CSG was applied using the number of samples of class i in the noisy dataset and 500 as the number of neighbors in the K-nearest algorithm for CIFAR-10n and 50 for CIFAR-100n. By applying the CSG metric to the data subsets, a similarity array was obtained that allowed the most confused samples to be detected according to the probability of each sample. This probability was used as a confidence score, which made it possible to order the samples from the most confused to the least confused.

3.3. Relabel

After sorting the samples of each class according to the confidence score (from highest to lowest), the first N samples of each class are selected and relabeled according to the clean labels contained in the CIFAR-10n dataset. In this way, for each value of N, a new version of the CIFAR-10n-worst dataset is obtained, where N samples of each class have been relabeled.

This process was applied by varying N from 200 to 3200 samples with increments of 200 samples in CIFAR-10n and varying N from 40 to 640 samples with increments of 40 samples in CIFAR-100n. After relabeling 3200 samples in CIFAR-10n and 640 samples in CIFAR-100n, the dataset is balanced, i.e., the number of samples in each class is equal to the number of classes in the original CIFAR-10 dataset.

3.4. Fine-Tuning

During the fine-tuning process, the pre-trained DenseNet121 model is used, using the weights obtained from its training on the ImageNet dataset. The top layers of the model are excluded to adapt it to the 10 classes of the CIFAR-10 dataset. Consequently, in DenseNet, a global average pooling layer is added, followed by a fully connected layer of 256 units with a dropout of 0.5 and a final fully connected layer where the number of units is equal to the number of classes in the dataset. The first 141 layers of the model are frozen, as they contain the general features learned during pre-training that are very useful for a wide range of tasks. Layers from the 141st onward are set as trainable, allowing the model to adapt and fine-tune its parameters according to the specific datasets being evaluated. In Xception, the top layer is adjusted through a global average pooling layer and a fully connected layer with the same number of units as the number of classes in the dataset, and the first 117 layers are frozen.

3.5. Evaluation

After fitting and training the DenseNet121-based models, inference is performed on the CIFAR-10/100 test data. This ensures that the model is evaluated on an independent dataset, providing a reliable assessment of its performance. Each result is consolidated by a 10-class (100-class in CIFAR-100n) confusion matrix, allowing different metrics to be calculated. In particular, the accuracy (7) and the F-score (8) are obtained.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$FScore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

3.6. The Experimental Environment

Google Colab, a cloud platform that provides high-performance computing resources, was used for sample correction. The development environment was configured in Python 3.11, taking advantage of specialized libraries such as spectral-metric (CSG) [28]. This environment can be run with basic resources such as a CPU (the CSG does not require a GPU) and the RAM available in the basic version (12.67 GB).

Google Colab Pro+ was used to train and evaluate the deep learning models. The development environment was configured in Python, taking advantage of specialized libraries such as Keras/TensorFlow. To speed up the training process, an A100 GPU was used, which significantly reduced the training time. In addition, 83.48 GB of RAM was available.

4. Results

The results after applying the proposed methodology are structured in terms of the balance of the number of samples per class, the overall performance metrics, the performance metrics per class, a multi-class scenario, and comparison with other methods.

4.1. Balance of the Number of Samples per Class

As mentioned above, due to the labeling process in CIFAR-10n-worst, each of the classes has a high level of noise, i.e., they contain a large number of images that do not correspond to their class, but also the number of images in each class varies with respect to the number of samples in the CIFAR-10 dataset.

By applying the proposed methodology, a certain number of erroneous samples is identified in each class and transferred to its corresponding class. Thus, each class not only sends images to other classes but also receives images from other classes. Since the number of images sent and received is not necessarily equal, the number of samples per class can both increase and decrease. In fact, the standard deviation in the number of images per class initially stays around 600 and begins to decrease from 1200 corrected samples per class until it reaches zero at 3200 corrected samples per class (i.e., the point at which the classes are balanced) (see Figure 4).

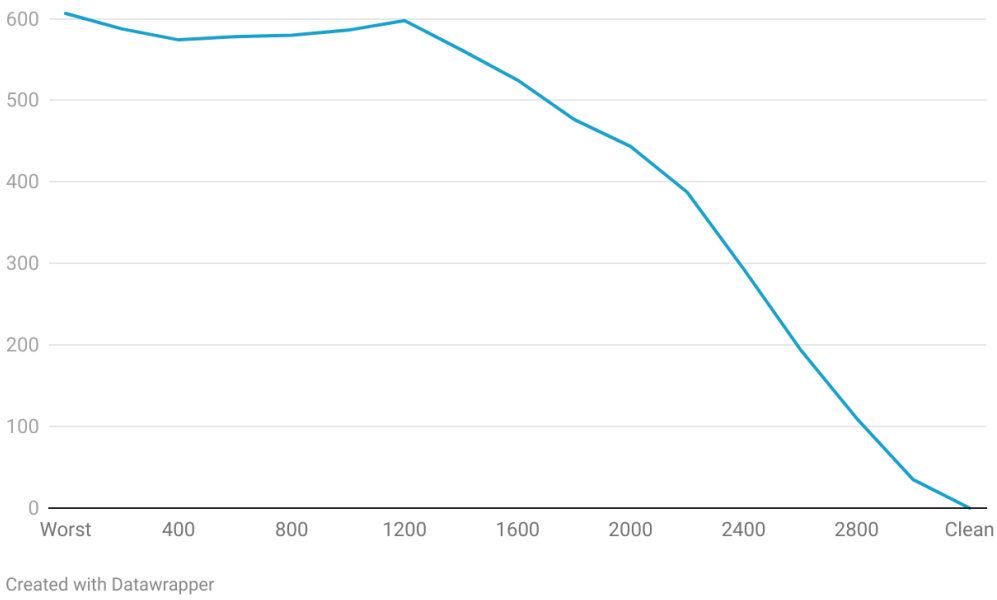


Figure 4. Standard deviation in the number of images per class as a function of the number of corrected samples.

The variation in the number of samples in each class is shown in Figure 5. In this case, we have classes that gradually increase in the number of samples (deer, frog), classes that gradually decrease in the number of samples (car, dog), and classes that decrease and then increase in the number of samples. In each case, once the relabeling process stabilizes, the dataset reaches equilibrium, with an equal number of samples per class.

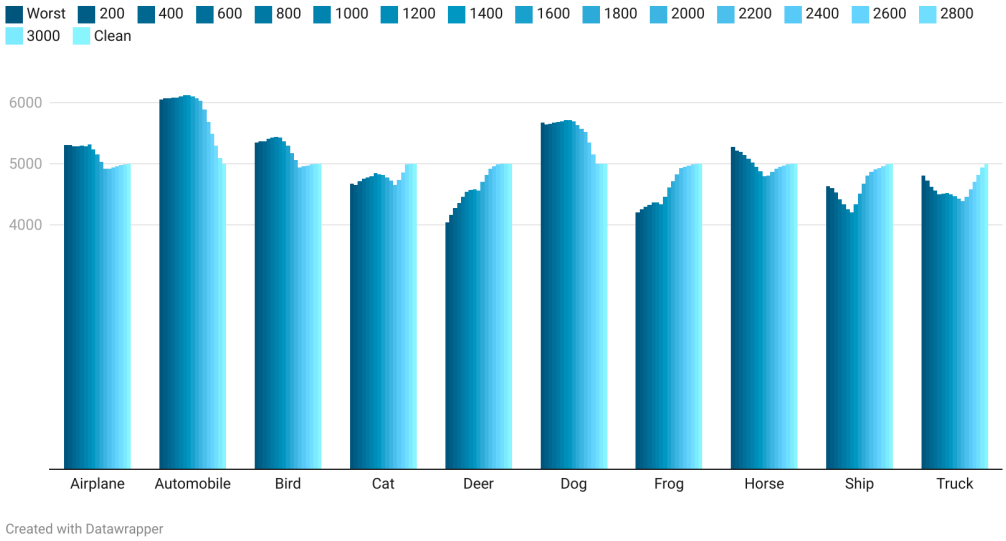


Figure 5. Number of samples per class as a function of the number of corrected samples.

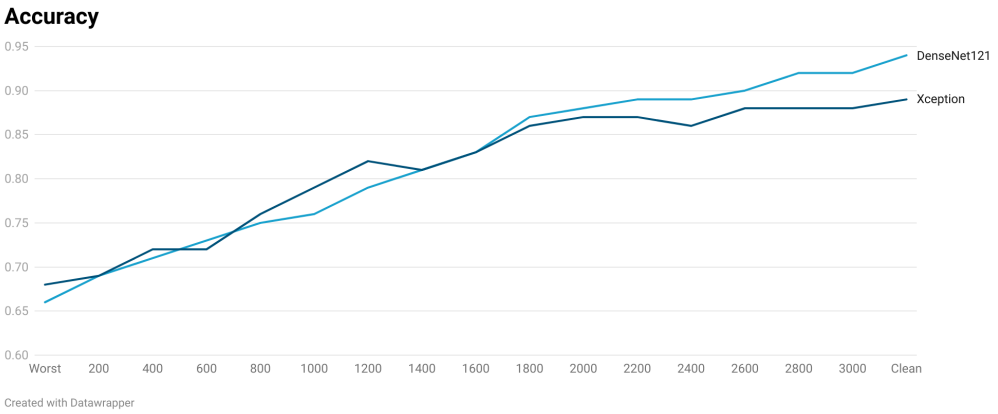
4.2. Overall Performance

In addition to the balance of the dataset, it is important to evaluate whether the versions of the dataset obtained after applying the proposed method have an advantage in terms of the deep learning model's performance.

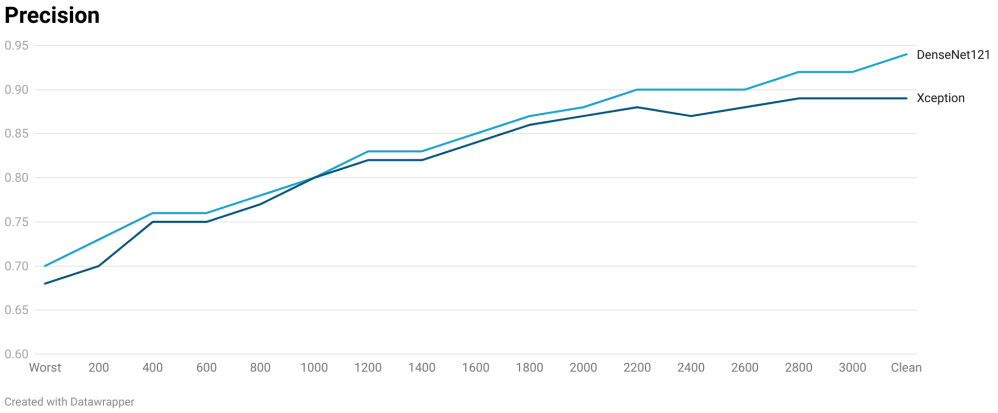
For this purpose, the models based on DenseNet121 and Xception were trained with each of the 17 available datasets (CIFAR-10n-worst: 200–3000 class-corrected images; CIFAR-10n-clean: 3200 class-corrected images). These results in terms of the accuracy and F-score are shown in Table 1 and Figure 6a–d.

Table 1. Average accuracy and F-score results for all classes as a function of the number of corrected samples in the CIFAR-10n dataset ("worst" version, ≈40% noise in the labels).

No. Samples Corrected	Accuracy (Average)	F-Score (Average)
Worst	0.66	0.66
200	0.69	0.68
400	0.71	0.71
600	0.73	0.73
800	0.75	0.75
1000	0.76	0.76
1200	0.79	0.80
1400	0.81	0.81
1600	0.83	0.83
1800	0.87	0.87
2000	0.88	0.88
2200	0.89	0.89
2400	0.89	0.89
2600	0.90	0.90
2800	0.92	0.92
3000	0.92	0.92
Clean	0.94	0.94

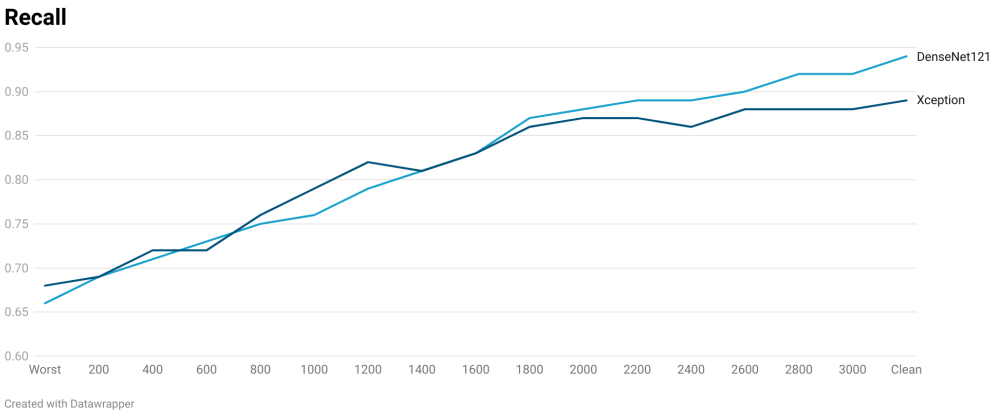


(**a**)

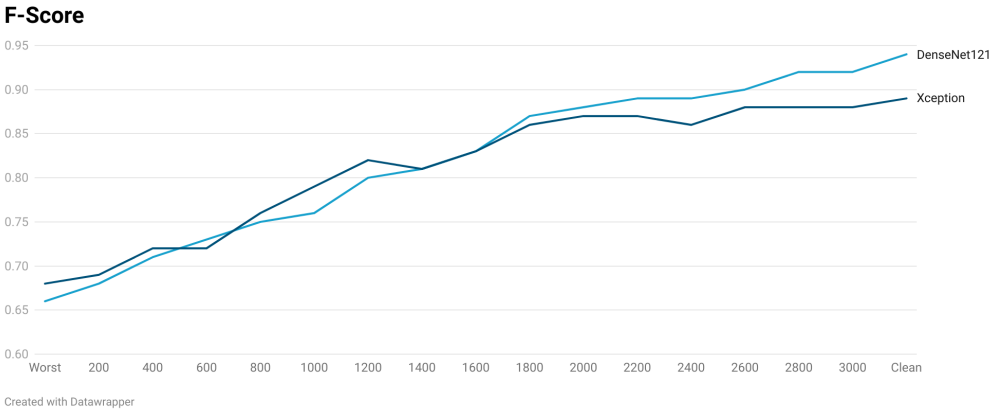


(**b**)

Figure 6. *Cont.*



(**c**)



(**d**)

Figure 6. Average evaluation metrics for all classes in the CIFAR-10 test data. Models: DenseNet121 and Xception fine-tuned with CIFAR-10n, increasing the number of corrected samples from the "worst" version (≈40% noise in the labels). Correction method: selection based on confidence score obtained with CSG. (**a**) Accuracy of two DL models trained with different versions of noisy data, evaluated on CIFAR-10 test data. (**b**) Precision of two DL models trained with different versions of noisy data, evaluated on CIFAR-10 test data. (**c**) Recall of two DL models trained with different versions of noisy data, evaluated on CIFAR-10 test data. (**d**) F-Score of two DL models trained with different versions of noisy data, evaluated on CIFAR-10 test data.

According to Figure 6a–d and Table 1, there is a strong correlation between the metrics. This is to be expected since both metrics measure the overall performance of the classifier. In this case, the values are practically equal, indicating a good balance between precision and recall on average. It is also observed that the accuracy and the average F-score improve significantly when the labels are corrected. Starting from a value of 0.66 in "worst", they reach a value of 0.92 with 3000 corrections. This is an improvement of 26 percentage points, which is significant.

Another aspect to consider has to do with the fact that most of the average improvement is concentrated in the early stages of correction, especially up to 2200 corrected samples. After this point, the improvement grows at a slower rate.

Finally, the performance on the clean dataset achieves an accuracy and an F-score of 0.94 with DenseNet and 0.89 with Xception. This gives us an idea of the standard performance achieved using DenseNet through transfer learning, which serves as a benchmark for evaluating the effectiveness of the correction after label identification and correction. It is observed that the model is quite close to this ideal performance with 3000 corrections.

To analyze the impact of the proposed method at the class level, Table 2 and Figure 7 are shown. As shown in Table 2, the CSG-based label correction results in a significant

improvement in the F-score for all classes. The absolute improvements vary between 0.19 and 0.31, which represents a significant relative increase between 25% and 51.79%. This confirms the effectiveness of the method in mitigating the effects of noise on the labels. In addition, considerable variability in the magnitude of the improvement between classes is observed.

Table 2. Improvement in F-score per class after label correction for CIFAR-10n.

Class	Initial F-Score ("Worst")	F-Score ("Clean")	Absolute Imp.	Relative Imp. (%)	Largest Imp. (ΔF-Score)	Range of Largest Imp. (Corrected Samples)
Airplane	0.69	0.93	0.24	34.78	0.17	200–600
Automobile	0.68	0.96	0.28	41.18	0.16	600–1400
Bird	0.66	0.90	0.24	36.36	0.12	600–1200
Cat	0.56	0.85	0.29	51.79	0.19	1800–2200
Deer	0.62	0.93	0.31	50.00	0.17	600–1400
Dog	0.58	0.87	0.29	50.00	0.18	1800–2200
Frog	0.76	0.95	0.19	25.00	0.14	1200–1800
Horse	0.68	0.94	0.26	38.24	0.15	1600–2000
Ship	0.74	0.95	0.21	28.38	0.17	400–800
Truck	0.66	0.95	0.29	43.94	0.18	1800–2200

In particular, the classes cat, deer, and dog show the largest relative improvements (about 50%), while frog shows the smallest (25%). This suggests that some classes have more initial noise than others and that the classifier has more difficulty learning the discriminative patterns in certain classes.

The column "Greatest Improvement (ΔF-score)" in Table 2 indicates the greatest increase in the F-score between two consecutive correction points, complemented by the column "Range of Greatest Improvement", which indicates the range of corrected samples in which this greatest improvement occurs. It is also observed that in most classes, the greatest improvement is centered between 600 and 2200 corrected samples. That is, most of the improvement is observed in the intermediate stages of correction (up to about 1600–2000 corrected samples). After this point, the improvement becomes more gradual, indicating a stabilization point. This suggests that correcting a large number of labels may not always be necessary and that a moderate number of strategic corrections (selected using the CSG) may be sufficient to achieve a significant improvement.

As can be seen in Figure 7, some classes exhibit certain behavior in the early stages of pattern correction. This is the case for the classes airplane, automobile, and truck. Airplane shows an initial drop in the F-score between "worst" and 200 corrected samples, followed by rapid recovery and subsequent stabilization. This could indicate an initial correction of samples that, although mislabeled, contained useful information for the classifier. Finally, the frog and boat classes show the highest initial performance ($\approx$0.75), suggesting that the noise level in these classes is lower compared to that in the other categories.

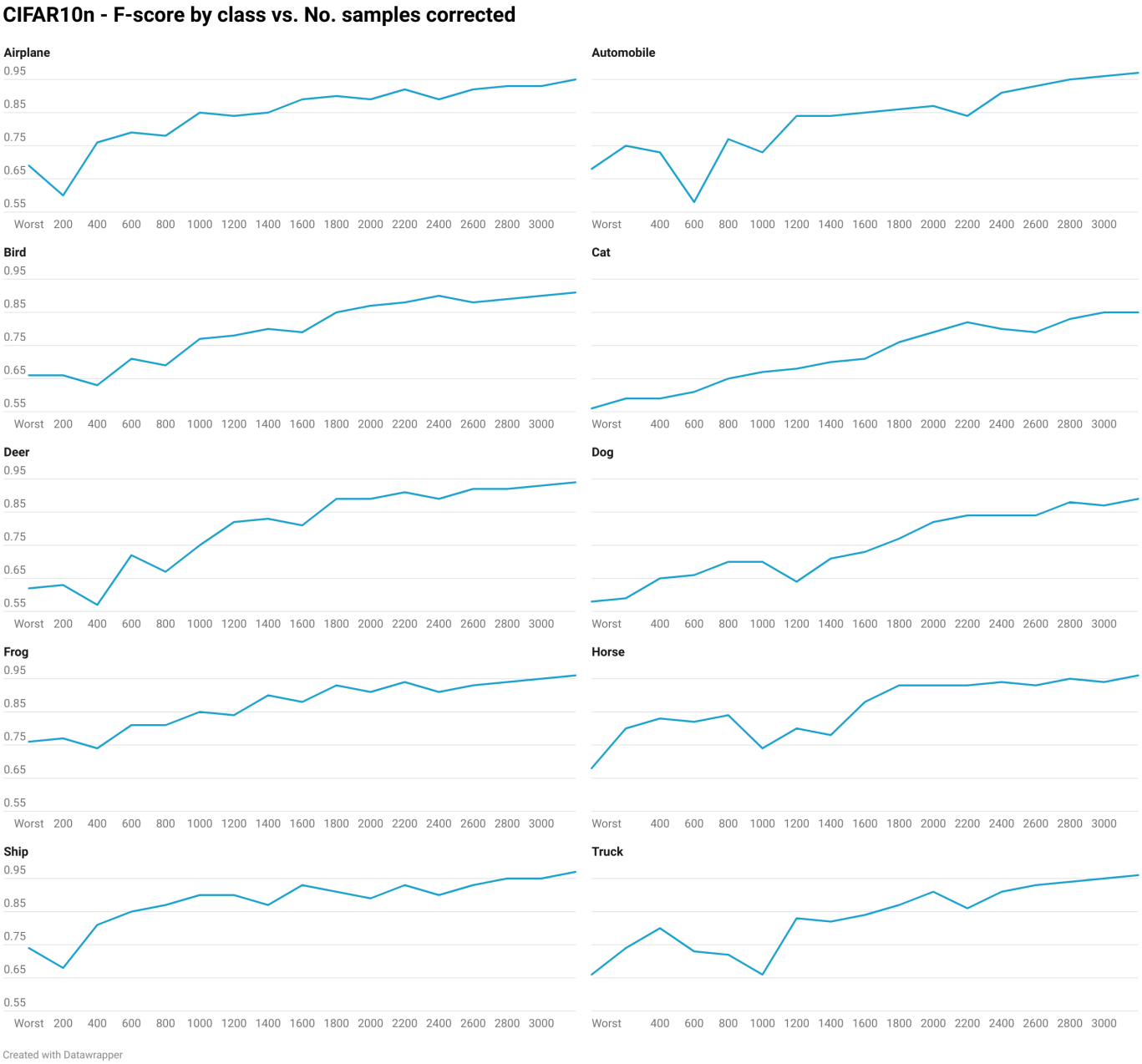


Figure 7. F-score per class in test data.

4.3. High-Dimensional Multi-Class Scenario

Taking into account that in practice there may be more complex problems with many more classes, the application of the proposed method was used in a scenario with 100 classes. In this case, the CIFAR-100-n dataset was used, whose samples were labeled with a noise level similar to that of CIFAR-10n—that is, ≈40% noise in the fine labels of the dataset (considering 100 classes).

The original version of the CIFAR-100n training subset has 500 images per class, while in CIFAR-100n, the number of images per class can vary from 199 to 865 samples in a class. Therefore, for CIFAR-100n, the number of corrected images per class was varied from 40 to 640 images, where all of the labels in the dataset were corrected.

Using this dataset and the corresponding sample-corrected versions of the data, the DenseNet121 model was fine-tuned, and the results were evaluated in a similar manner to the process for CIFAR-10n. These results are shown in Figure 8.

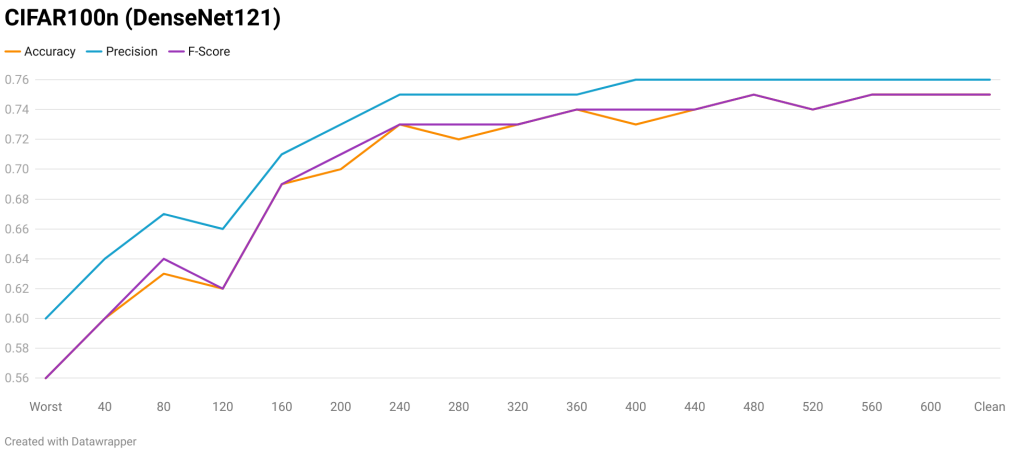


Figure 8. Average evaluation metrics for all classes in the CIFAR-100 test data. Model: DenseNet121 fine-tuned with CIFAR-100n, increasing the number of corrected samples from the noisy version (≈40% noise in the fine labels). Correction method: selection based on confidence score obtained with CSG.

According to the results shown in Figure 8, the model performance improves as the number of corrected images per class increases. The greatest impact is seen when the correction is performed after relabeling the first samples identified by the proposed method (40–240 samples per class). On the contrary, increasing the number of relabeled samples per class has no significant impact for higher values (280–600).

Finally, two of the advantages of the CSG-based methodology are that this metric has been extensively validated on diverse datasets and that its performance is closely related to the generalization of CNNs. Its performance has been tested on low-resolution grayscale image datasets such as MNIST and notMNIST, low-resolution color datasets such as CIFAR-10, and medium-resolution image datasets such as STL-10 or CompCars. Similarly, the application results for this metric have been contrasted with the generalized performance of models such as ResNet-50 or Xception. In any case, it may be appropriate to extend the validation of the proposed methodology to datasets with higher-resolution images or with a larger number of classes [19].

4.4. Comparison with the Other Methods

To put the performance of the models trained with CIFAR-10/100n into context, the results obtained are compared with the results presented in the original CIFAR-10/100n article. In this case, the three best results from this evaluation are included with respect to the best results of the proposed methodology. These values are presented in Table 3.

Table 3. Performance comparison of the proposed methodology with respect to the results presented in [27] for CIFAR-10n and CIFAR-100n.

Dataset	Method	Model	Accuracy
CIFAR-10n	[15]	ResNet-34	0.926
	[16]	ResNet-34	0.917
	[17]	ResNet-34	0.927
	Ours	DenseNet121	0.942
	Ours	Xception	0.895
CIFAR-100n	[18]	ResNet-34	0.667
	[29]	ResNet-34	0.711
	[17]	ResNet-34	0.704
	Ours	DenseNet121	0.748

It is important to note that these models were designed to train models in the presence of noisy labels, while the proposed methodology focuses on the identification of samples with the possible presence of noise in the dataset.

5. Conclusions

The results demonstrate the effectiveness of the proposed method based on the confidence score and the CSG for mitigating the impact of label noise on the CIFAR10n/100n datasets. A significant improvement in the performance of the DL-based classifier with fine-tuning is observed, both at the class level and on average, as erroneous labels are corrected. The quantitative analysis shows that most of the improvement is obtained with an intermediate number of corrections, which could be explored in further work. Furthermore, the performance obtained with 240 corrections in CIFAR-100n is significantly close to the performance obtained with clean data, confirming the effectiveness of this method. However, the variability in the improvement between classes highlights the need to consider the specific characteristics of each category.

Author Contributions: Conceptualization: C.G.-S. and D.R. Data curation: C.G.-S. Formal analysis: D.R. and E.M.-A. Investigation: C.G.-S. and D.R. Methodology: C.G.-S. and D.R. Resources: D.R. Supervision: D.R. and E.M.-A. Validation: D.R. and E.M.-A. Visualization: D.R. and E.M.-A. Writing—original draft: C.G.-S. Writing—review and editing: D.R. and E.M.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by "Universidad Militar Nueva Granada-Vicerrectoría de Investigaciones" under the grant INV-ING-3946 of 2024.

Data Availability Statement: The image datasets used in this work are publicly available from CIFAR10/100 [30], https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 1 August 2024), and CIFAR10n/100n [27], https://paperswithcode.com/dataset/cifar-10n (accessed on 1 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CSG	Cumulative Spectral Gradient
CNN	Convolutional Neural Network

References

1. Nguyen, G.; Dlugolinsky, S.; Bobák, M.; Tran, V.; López García, Á.; Heredia, I.; Malík, P.; Hluchỳ, L. Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artif. Intell. Rev.* **2019**, *52*, 77–124. [CrossRef]
2. Fan, C.; Guo, D.; Wang, Z.; Wang, M. Multi-Objective Convex Quantization for Efficient Model Compression. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *47*, 2313–2329. [CrossRef] [PubMed]
3. Northcutt, C.G.; Athalye, A.; Mueller, J. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. *arXiv* **2021**, arXiv:2103.14749.
4. Northcutt, C.G.; Jiang, L.; Chuang, I.L. Confident Learning: Estimating Uncertainty in Dataset Labels. *arXiv* **2022**, arXiv:1911.00068.
5. Muller, N.M.; Markert, K. Identifying Mislabeled Instances in Classification Datasets. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8. [CrossRef]
6. Zhang, Y.; Li, B.; Ling, Z.; Zhou, F. Mitigating label bias in machine learning: Fairness through confident learning. *Proc. Aaai Conf. Artif. Intell.* **2024**, *38*, 16917–16925.
7. Hu, Z.; Zhang, Z.; Liu, Q.; Bi, H.; Huang, Z.; Mao, Q.; Gao, W.; Feng, W. Mitigating Bias with Incomplete Sensitive Labels: A Confidence-Based Randomization Framework. In *International Conference on Database Systems for Advanced Applications*; Springer: Singapore, 2024; pp. 139–155.
8. Ju, L.; Wang, X.; Wang, L.; Mahapatra, D.; Zhao, X.; Zhou, Q.; Liu, T.; Ge, Z. Improving medical images classification with label noise using dual-uncertainty estimation. *IEEE Trans. Med. Imaging* **2022**, *41*, 1533–1546. [CrossRef] [PubMed]

9. Wang, Z.; Voiculescu, I. Dealing with unreliable annotations: A noise-robust network for semantic segmentation through a transformer-improved encoder and convolution decoder. *Appl. Sci.* **2023**, *13*, 7966. [CrossRef]
10. Shi, J.; Zhang, K.; Guo, C.; Yang, Y.; Xu, Y.; Wu, J. A survey of label-noise deep learning for medical image analysis. *Med. Image Anal.* **2024**, *95*, 103166. [PubMed]
11. Burgert, T.; Ravanbakhsh, M.; Demir, B. On the effects of different types of label noise in multi-label remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5413713.
12. Pelletier, C.; Valero, S.; Inglada, J.; Champion, N.; Marais Sicre, C.; Dedieu, G. Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sens.* **2017**, *9*, 173. [CrossRef]
13. Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral image classification in the presence of noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 851–865. [CrossRef]
14. Jiang, H.; Nachum, O. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*; PMLR: Cambridge, MA, USA, 2020; pp. 702–712.
15. Li, J.; Socher, R.; Hoi, S.C. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv* **2020**, arXiv:2002.07394.
16. Cheng, H.; Zhu, Z.; Li, X.; Gong, Y.; Sun, X.; Liu, Y. Learning with instance-dependent label noise: A sample sieve approach. *arXiv* **2020**, arXiv:2010.02347.
17. Bai, Y.; Yang, E.; Han, B.; Yang, Y.; Li, J.; Mao, Y.; Niu, G.; Liu, T. Understanding and improving early stopping for learning with noisy labels. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24392–24403.
18. Liu, S.; Niles-Weed, J.; Razavian, N.; Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 20331–20342.
19. Branchaud-Charron, F.; Achkar, A.; Jodoin, P.M. Spectral metric for dataset complexity assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3215–3224.
20. Lozano, J.; Renza, D.; Ballesteros, D. Evaluation of pruning according to the complexity of the image dataset. *J. Inf. Hiding Multimed. Signal Process.* **2024**, *15*, 53–62.
21. Li, G.; Togo, R.; Ogawa, T.; Haseyama, M. Dataset complexity assessment based on cumulative maximum scaled area under Laplacian spectrum. *Multimed. Tools Appl.* **2022**, *81*, 32287–32303.
22. Renza, D.; Moya-Albor, E.; Chavarro, A. Adversarial Validation in Image Classification Datasets by Means of Cumulative Spectral Gradient. *Algorithms* **2024**, *17*, 531. [CrossRef]
23. Aguilera-González, S.; Renza, D.; Moya-Albor, E. Evaluation of Dataset Distribution in Biomedical Image Classification Against Image Acquisition Distortions. In Proceedings of the 2024 20th International Symposium on Medical Information Processing and Analysis (SIPAIM), Antigua, Guatemala, 13–15 November 2024; pp. 1–6.
24. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
25. Saravia, E. ML Visuals. 2021. Available online: https://github.com/dair-ai/ml-visuals (accessed on 14 February 2025).
26. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
27. Wei, J.; Zhu, Z.; Cheng, H.; Liu, T.; Niu, G.; Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv* **2021**, arXiv:2110.12088.
28. Branchaud-Charron, F.; Achkar, A.; Jodoin, P.M. Spectral Metric. 2019. Available online: https://github.com/Dref360/spectral-metric (accessed on 14 February 2025).
29. Wei, J.; Liu, H.; Liu, T.; Niu, G.; Liu, Y. Understanding Generalized Label Smoothing When Learning with Noisy Labels. 2021. Available online: https://openreview.net/forum?id=UQQgMRq58O (accessed on 14 February 2025).
30. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf (accessed on 20 March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.