International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 3, June 2025, pp. 2951~2964
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i3.pp2951-2964

Enhancing training performance for small models using data-centric approaches

Reda A. El-Khoribi[1], Eid Emary[1,2], Amr Essam Hassan[1]
[1]Department of Information Technology, Faculty of Computers and Artificial Intelligence, University of Cairo, Cairo, Egypt
[2]Faculty of Computing Studies, Arab Open University, Cairo, Egypt

Article Info

Article history:

Received Jul 15, 2024
Revised Dec 19, 2024
Accepted Jan 16, 2025

Keywords:

Computer vision
Data-centric
Deep learning
Generative adversarial network
Model-centric

ABSTRACT

In this work, we propose a new system to improve the performance of classification models by applying data-centric principles. The system optimizes datasets by removing poor-quality samples and generating high-quality synthetic data. We tested the system on various classification models and datasets, measuring its performance with accuracy, precision, recall, and F1-score. The results showed significant improvements in classification performance, highlighting the effectiveness of this data-centric approach. While the scalability to large-scale datasets is still an open question, it offers great potential for future research. This approach could be valuable in critical areas like healthcare, finance, and autonomous systems, where high-quality data is crucial. Future work could explore advanced data augmentation, adapting the system for different data types like text and time-series, and extending it to semi-supervised and unsupervised learning. Our findings emphasize the importance of data quality in achieving better model performance, often overlooked in favor of model architecture. By advancing data-centric artificial intelligence (AI), this work offers a practical framework for researchers and practitioners to optimize datasets and improve machine learning systems.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Amr Essam Hassan
Department of Information Technology, Faculty of Computers and Artificial Intelligence, University of Cairo
Cairo, Egypt
Email: amr.essam.hassan@outlook.com

1. INTRODUCTION

Deep learning has revolutionized artificial intelligence (AI), driving advancements in domains such as medical diagnosis, autonomous systems, and large-scale decision-making [1]. These developments have significantly improved efficiency, accuracy, and innovation across various industries. However, challenges remain, particularly regarding the availability and quality of datasets crucial for training effective models [2]–[4]. The scarcity of reliable labeled data and the high acquisition costs continue to hinder AI development. For example, medical imaging datasets often contain mislabeled samples, low-resolution images, or insufficient representations of rare diseases, all of which degrade model performance [5]. Although data augmentation techniques, such as flipping and noise injection, are commonly applied, they often introduce biases or fail to fully address dataset deficiencies [6], [7]. As models become more complex, their performance increasingly depends on high-quality datasets rather than solely on architectural innovations [3], [4]. This shift has given rise to data-centric AI, which focuses on optimizing datasets rather than exclusively enhancing model architectures [2], [3]. Data-centric methodologies prioritize refining datasets by addressing label noise, class imbalance, and irrelevant data points [4]–[8].

Andrew Ng's data-centric AI competition underscored the importance of dataset quality in achieving superior model performance, even with smaller datasets [1]. By systematically enhancing data quality, researchers have demonstrated notable improvements in model robustness, generalization, and efficiency across diverse applications [2]–[7]. This competition highlighted the often-overlooked role of high-quality data in driving performance gains, even when using simpler or smaller models, reinforcing the shift toward data-centric AI practices. Building upon these insights, this study introduces a data-centric algorithm designed to improve datasets for training robust deep neural networks (DNNs). The algorithm employs techniques to identify and remove noisy or mislabeled samples, which can distort model learning and reduce performance. Additionally, it incorporates high-quality synthetic data generated using generative models such as generative adversarial networks (GANs) [9]. These GAN-generated samples address issues like class imbalance and dataset sparsity, providing the model with a more comprehensive and representative training set. This approach proves particularly valuable in high-stakes domains like healthcare and finance, where the accuracy and reliability of predictive models are critical for informed decision-making [4]–[8], [10]. By demonstrating that prioritizing dataset quality can significantly improve performance, this research contributes to the growing body of evidence supporting data-centric AI as a complementary methodology to model-centric approaches. The findings suggest that an effective balance between these two paradigms leads to better performance and more stable outcomes. The proposed framework, therefore, emphasizes the importance of integrating both perspectives to advance machine learning systems, ensuring that models can learn from cleaner, more meaningful data while leveraging optimized architectures for enhanced performance.

2. RELATED WORKS

Significant prior research has advanced data-centric AI by addressing dataset quality issues. Study [1] demonstrated that variations in dataset size, labeling quality, and train-test splits substantially impact model performance, underscoring the importance of data-centric methodologies. Similarly, Sambasivan *et al.* [2] emphasized the ripple effects of labeling errors, advocating for systematic data quality improvements.

Northcutt *et al.* [11] identified pervasive labeling errors in widely used datasets, reinforcing the need for data-centric strategies. Polyzotis and Zaharia [4] proposed end-to-end version tracking and actionable monitoring for managing dynamic. datasets in production systems. Hamid [12] highlighted the applicability of data-centric AI in Industry 4.0 by enhancing data quality for robust industrial automation.

In natural language processing (NLP), Xu *et al.* [3] introduced the data CLUE benchmark, showing simple yet effective strategies for improving data quality. Seedat *et al.* [13] proposed DC-check, a checklist for systematically evaluating data-centric reliability. Seedat *et al.* [14] also emphasized dataset transparency with datasheets for datasets, a foundational tool in high-stakes applications.

Motamedi *et al.* [6] demonstrated data quality enhancements with GAN-generated samples, achieving improved accuracy while reducing dataset size. Ma *et al.* [15] reviewed data-centric AI's role in addressing labeling errors and class imbalances. Shankar and Evans [16] identified pitfalls in dataset construction, proposing best practices for reliability and fairness.

Zha *et al.* [5] introduced a framework integrating statistical measures to ensure dataset reliability. These studies highlight the transformative potential of data-centric AI in addressing data quality issues, forming. The foundation for the dataset optimization approach presented in this work.

3. METHOD

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given input data example. To construct such a predictive model, we define loss functions, set hyperparameters for the model, and, given a training dataset, optimize the model parameters to minimize the loss function, as expressed in (1).

$$\min_{\phi,\gamma,\theta} \text{loss}\left(f(X,Y)\right) \tag{1}$$

The targeted loss function, denoted as *loss*, plays a crucial role in model optimization, where *f* is the mapping model, φ represents the hyperparameters, θ the model parameters, and γ the given dataset. Recent advancements in deep learning have established it as a powerful tool for learning hierarchical representations from large datasets, driving breakthroughs in domains like image recognition, natural language processing, and autonomous systems [1]–[8], [10]–[17]. To optimize performance, various loss functions have been developed based on task requirements, such as mean squared error for regression and cross-entropy loss for classification [18], [19].

3.1. Regression losses

Regression losses quantify discrepancies between predicted and actual continuous values:

a. Mean squared error (MSE): Penalizes large errors more heavily, making it sensitive to outliers [20].

$$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (2)$$

b. Mean absolute error (L1): Measures absolute differences, less sensitive to outliers [1].

$$L_{\text{L1}} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (3)$$

c. Huber loss: Combines MSE for small errors and L1 for large errors, offering robustness [21].

$$L_{Huber} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & if\,|y_i - \hat{y}_i| \le \delta, \\ \delta|y_i - \hat{y}_i|, & otherwise. \end{cases} \quad (4)$$

3.2. Classification losses

Classification losses evaluate model performance in classification tasks by measuring discrepancies between predicted and actual class labels:

a. Binary cross-entropy (BCE): Used for binary classification by measuring the difference between probability distributions [20].

$$L_{\text{BCE}} = -\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (5)$$

b. Categorical cross-entropy (CCE): Extends BCE to multi-class classification by summing the cross-entropy for all classes [20].

$$L_{\text{CCE}} = -\sum_{i=1}^{n}\sum_{c=1}^{C} y_{ic} \log(\widehat{y_{ic}}) \quad (6)$$

c. Focal loss (FL): Addresses class imbalance by down-weighting easy examples and emphasizing hard ones [22].

$$L_{\text{FL}} = -\alpha(1 - \hat{y}_i)^{\gamma} \log(\hat{y}_i) \quad (7)$$

3.3. Other losses

Specialized loss functions address unique challenges in training models:

a. Kullback-Leibler divergence (KL): Measures the divergence between two probability distributions, commonly used in probabilistic models [17].

$$L_{\text{KL}} = \sum_{i=1}^{n} y_i \log\frac{y_i}{\hat{y}_i} \quad (8)$$

b. Hinge loss: Used in support vector machines (SVMs) to maximize class margins [23].

$$L_{\text{Hinge}} = \max(0, 1 - y_i\hat{y}_i) \quad (9)$$

c. Total variation (TV): Promotes smoothness in images or signals, reducing noise [7].

$$L_{\text{TV}} = \sum_{i,j}\left(\left|\left|x_{i+1,j} - x_{i,j}\right|\right| + \left|\left|x_{i,j+1} - x_{i,j}\right|\right|\right) \quad (10)$$

3.4. Hyperparameters

Hyperparameters are predefined parameters critical to the performance of machine learning models. Their selection significantly impacts model training and outcomes.

a. Learning rate (η): Determines the step size for gradient descent. Smaller η ensures convergence but slows training, while larger η may cause instability [17].

$$\theta \coloneqq \theta - \eta\nabla_{\theta}J(\theta) \quad (11)$$

b. Batch size: Specifies the number of training examples per iteration. Smaller batches reduce overfitting but increase gradient noise [17]–[25].

c. Depth and width: The number of layers and neurons per layer impact the model's representational capacity.
d. Excessive layers or neurons may lead to overfitting and inefficiencies [17]–[25], [26] [27].
e. Activation functions: Add non-linearity to networks, enabling complex pattern learning. Common choices include ReLU, sigmoid, and tanh [17]–[25], [26]–[30].
f. Regularization parameters: Prevent overfitting by penalizing large weights. L1 promotes sparsity; L2 reduces variance [28]–[32].

3.5. Dataset optimization

Optimizing the dataset is a critical aspect of this work. Operations such as pruning training points, inserting newly generated data points, and weighting specific points in the loss function are performed to enhance dataset quality. These optimizations aim to improve the representativeness, balance, and cleanliness of the training dataset, leading to more robust and accurate machine learning models.

Before implementing dataset optimization, diagnosing potential issues within the dataset is necessary to identify appropriate remedies. A straightforward approach to diagnosing dataset problems involves training a basic model on the dataset and analyzing its performance to detect noisy or mislabeled samples. Once identified, these problematic samples can either be corrected or removed.

The dataset optimization process includes two primary operations:

a. Sample pruning: Removing noisy or mislabeled samples from the dataset.
b. Sample insertion: Augmenting the dataset with high-quality synthetic samples to address class imbalance or under-represented feature.

We begin by training a baseline model on the original dataset, and based on its outputs, we apply these enhancement operations.

3.5.1. Sample pruning

Sample pruning involves identifying and removing noisy or problematic samples from the dataset, as shown in Figure 1, These issues may include mislabeled examples, ambiguous data points, or samples with conflicting features. This process typically relies on techniques such as confident learning, which estimates the joint distribution of noisy and clean labels to detect and address errors systematically. The steps involved in sample pruning are as follows: i) collect or select a representative dataset; ii) train a baseline model on the dataset to establish a performance benchmark; iii) use cross-validation on the training data to identify noisy samples; iv) apply techniques such as label errors and confident learning to detect mislabeled or ambiguous data points; and v) remove or reweight the problematic samples to obtain a clean dataset for training.

3.5.2. Sample insertion

Following the pruning of noisy samples, as shown in Figure 2, new samples are generated to enrich the dataset. This process involves identifying weak classes, addressing naming errors, and training a GAN to generate new instances for the underrepresented class. This step ensures better class representation and helps the model learn more effectively from the available data. The steps for implementing sample insertion are as follows: i) Collect or select a representative dataset to serve as the foundation for training; ii) Train the dataset on a baseline model and calculate the initial accuracy to establish performance benchmarks; iii) Identify the weak class by analyzing the accuracy results and determining which classes underperform; iv) Train a GAN model specifically for the weak class to generate high-quality synthetic instances that improve class balance and overall model performance [32]; and v) Add the generated instances to the existing dataset to improve representation and classification accuracy.

After noisy samples are pruned in Phase 1, the remaining dataset is enriched by generating new synthetic samples. Weak classes are identified and addressed by generating instances using clean data to train a GAN model. These new samples enhance the dataset's representation and classification accuracy. The GAN model's optimization problem is defined as [32].

$$\min_G \max_D V(D,G) = E_{x\sim p_{\text{dt}}(x)}[\log D(x)] + E_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (12)$$

The generator's loss function minimizes the maximum value of the discriminator's value function by generating samples likely to be misclassified by the discriminator [32].

$$\min_D \max_G V(D,G) = E_{x\sim p_{\text{dt}}(x)}[\log D(x)] + E_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (13)$$

The discriminator's loss function maximizes its value function while minimizing the generator's value function, enabling it to correctly classify both real and generated samples. The resulting enhanced dataset is:

$$\hat{\phi} = \phi^{\text{selected}} \cup \phi^{\text{generated}} \quad (14)$$

Here, ϕ^{selected} represents the pruned dataset, and $\phi^{\text{generated}}$ represents newly generated samples. The optimization ensures that a model f trained on $\hat{\phi}$ outperforms the original dataset ϕ.

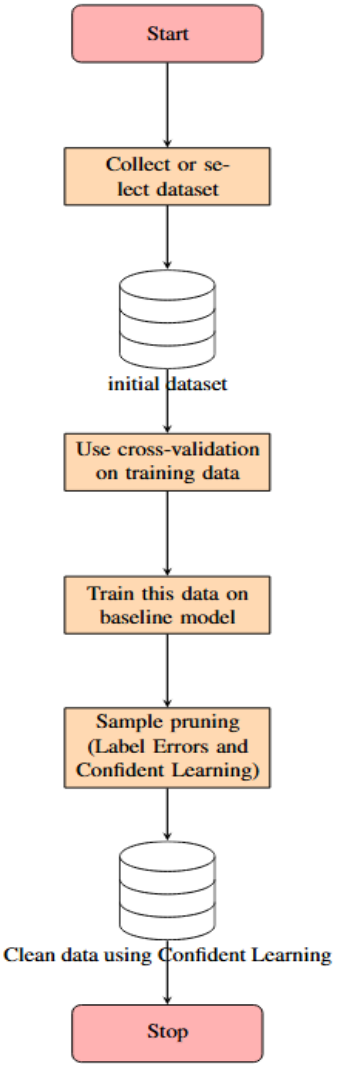

Figure 1. Sample pruning subsystem

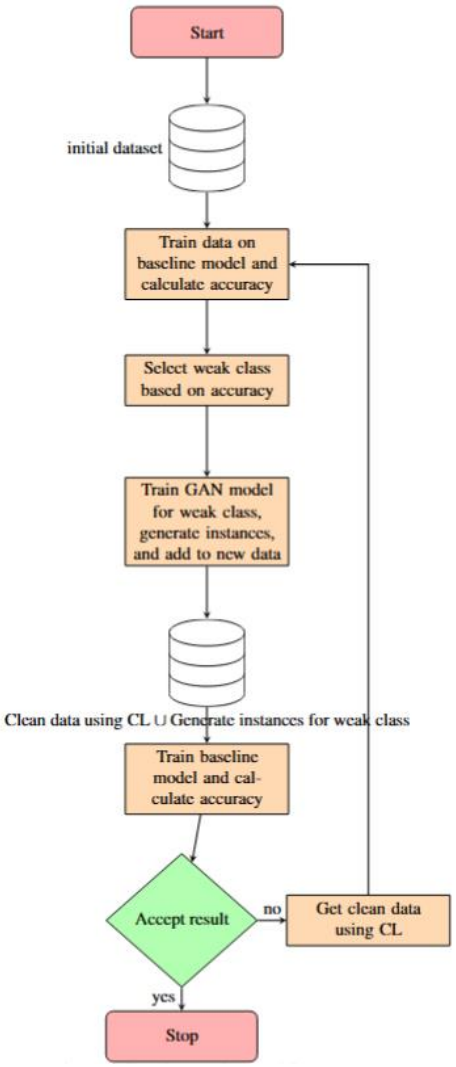

Figure 2. Sample insertion workflow

3.6. Experimental setup

We validated the proposed data-centric methods using the CIFAR-10 dataset [9], which includes 60,000 32×32 color images across 10 classes (50,000 for training and 10,000 for testing). Preprocessing addressed label noise and low-resolution samples via pruning and GAN-based augmentation.

a. Baseline models:
   - ResNet-18 [33]: An 18-layer CNN leveraging residual connections for gradient flow, pre-trained on ImageNet.
   - InceptionV3 [34]: A deep CNN with factorized convolutions and auxiliary classifiers for improved accuracy.

b. Evaluation metrics: Performance was assessed using accuracy, precision, recall, F1-score, and standard deviation.

c. Training configuration: Training employed the Adam optimizer [30] with a learning rate of 1×10^{-4} batch size of 128, 50 epochs, and standard augmentations (random cropping, flipping, normalization). Experiments were performed on an NVIDIA Tesla V100 GPU (32 GB).

d. Proposed method setup:
   - Sample pruning: confident learning [35] identified and removed noisy samples based on label uncertainty.
   - Sample insertion: d GAN [32] generated synthetic samples for underrepresented classes using a standard GAN architecture with a 100-dimensional latent vector. Training consisted of 10,000 iterations, with visual inspection of generated samples for quality assurance.
e. Baseline comparison:
   - Original dataset: baseline models trained on unaltered CIFAR-10.
   - Pruned dataset: models trained on high-quality samples post-pruning.
   - Enhanced dataset: models trained on pruned samples augmented with GAN-generated instances.

4. RESULTS AND DISCUSSION

In this section, we present the outcomes of our experiments and analyses aimed at enhancing the training performance of small models using data-centric approaches [1], [2]. The focus is on two deep classification models, InceptionV3 and ResNet18, which serve as our baseline models. We detail the performance improvements achieved through various stages of data optimization, including pruning and augmentation with generated samples. The evaluation criteria used for assessing model performance include accuracy, precision, recall, F1-score, and standard deviation [3].

4.1. Adopted models

Generally, any mapping model f can be adopted as a baseline for testing our proposals for data-centric enhancement. In this study, we used two models as baselines: InceptionV3 and ResNet18, which are recent examples of deep classification models [33], [34].

a. InceptionV3: This image recognition model has demonstrated accuracy greater than 78.1% on the ImageNet dataset. It integrates symmetric and asymmetric building blocks such as convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is applied extensively throughout the model, enhancing its robustness, while loss computation is performed using Softmax [27].
b. ResNet18: A convolutional neural network that is 18 layers deep, ResNet-18 has been pre-trained on over a million images from the ImageNet database. It achieves classification across 1,000 categories, including various objects and animals, utilizing residual learning to enhance gradient flow through its layers [28].

4.2. Dataset

We adopted the public CIFAR-10 dataset to validate the optimization methods applied to these baseline models. The CIFAR-10 dataset consists of 60,000 32×32 color images divided into 10 mutually exclusive classes, each containing 6,000 images [9]. The dataset is further split into 50,000 training images and 10,000 test images. The dataset's classes are designed to be entirely distinct. For example, the “automobile” class includes sedans and SUVs, while the “truck” class exclusively covers large trucks, with no overlap [15]. Figure 3 highlights common label errors found in the dataset, which can adversely affect model training. These errors underscore the importance of data cleaning to ensure accurate and reliable learning outcomes [8]. Additionally, several images in the dataset suffer from poor resolution, further hampering effective learning by the models. Low-resolution images often lack the necessary detail for feature extraction, reducing the model's ability to distinguish between similar classes. Figure 4 illustrates such low-resolution images, highlighting the importance of optimizing dataset quality through resolution enhancement techniques to achieve better performance [11].

Figure 3. Examples of label error samples. These types of errors emphasize the necessity of data cleaning to ensure accurate model training

Figure 4. Examples of low-resolution images in the dataset. Addressing these issues through data cleaning and enhancement is crucial for improving model performance

4.3. Evaluation criteria

To assess the performance of the models, several standard classification metrics were used. These metrics capture different aspects of model accuracy, robustness, and reliability, aligning with best practices in machine learning evaluation [20]–[28].

a. Accuracy: This metric measures the proportion of correctly classified instances out of the total number of instances. It is formally expressed as (15) [16]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

b. Precision: Precision focuses on the accuracy of the positive predictions. It is defined as (16) [16]:

$$Precision = \frac{TP}{TP+FP} \quad (16)$$

c. Recall: Recall, also known as sensitivity, evaluates the model's ability to correctly identify all relevant instances. It is calculated as (17) [16]:

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

d. F1-score: The F1-score combines precision and recall into a single harmonic mean, providing a balanced metric for evaluating performance on datasets with class imbalances [16]:

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (18)$$

e. Significance test (t-test): A t-test was employed to compare performance metrics before and after applying the proposed data optimization techniques. The test assesses whether the observed differences are statistically significant. It is calculated using:

$$t_i^{a,b} = \frac{m_i^a - m_i^b}{\sqrt{S^2\left(\frac{1}{N}+\frac{1}{N}\right)}},$$

where (a) and (b) represent the two systems under comparison, (i) denotes the performance metric, (N) is the sample size, and (S^2) is the pooled variance [13]–[19].

f. Standard deviation: This metric measures the variability of the performance metrics across different classes, highlighting discrepancies or inconsistencies in classification results. Reducing standard deviation is crucial for ensuring balanced performance across all classes [2]–[4].

4.4. ResNet18 performance evaluation

The impact of data cleaning and optimization was analyzed using the ResNet18 and InceptionV3 models as the baseline architectures. Figure 5 provides a comprehensive overview of the problematic samples within the dataset, including label errors and low-resolution images. These issues were addressed through rigorous data preprocessing [4]–[8], [10]–[17], [18]–[20].

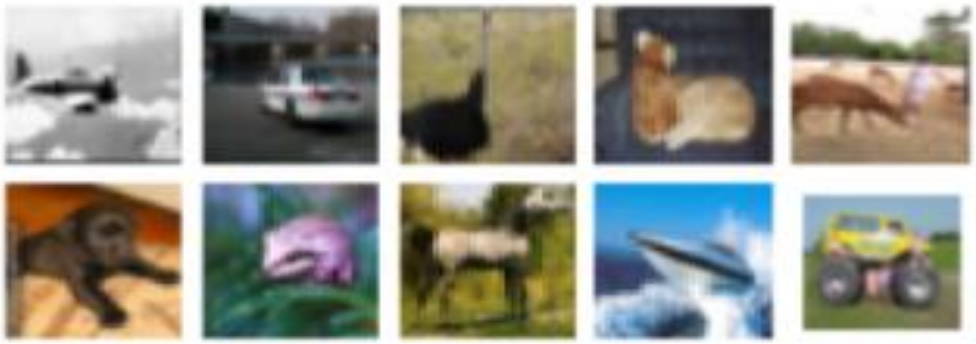

Figure 5. Examples of label errors and low-resolution images in the dataset. Addressing these issues was critical for optimizing model performance [4]–[8], [10]–[17], [18]–[21]

The performance metrics, including precision, recall, F1-score, and accuracy, were measured at different stages of the optimization process. Tables 1, 2, 3, and 4 summarize the results for ResNet18. The first column represents the model's baseline performance on the original dataset, while the second column reflects the performance after data cleaning. The final column demonstrates the enhancements achieved by incorporating generated samples using a GAN [28], [29].

Table 1 highlights the improvements in precision observed after each stage of dataset optimization. While pruning initially reduces the number of training samples, it helps eliminate noisy data, resulting in better precision for some classes. Incorporating generated samples further enhances precision, particularly for underrepresented or misclassified classes, such as “cat” and “dog” [28], [29].

Table 1. Precision comparison (ResNet18). This table compares the precision of ResNet18 across different stages: baseline, after pruning, and after incorporating generated samples

Class	Baseline	After Pruning	With GAN
Airplane	0.869	**0.887**	0.822
Automobile	0.929	**0.984**	0.919
Bird	**0.942**	0.950	0.856
Cat	0.786	0.659	**0.843**
Deer	0.819	**0.954**	0.877
Dog	**0.870**	0.627	0.876
Frog	0.877	0.861	**0.899**
Horse	0.917	0.843	**0.963**
Ship	0.923	0.941	**0.974**
Truck	0.979	0.747	0.955
Average	0.891	0.845	**0.898**
Std Dev	0.059	0.127	**0.053**

Table 2 demonstrates how recall improves after data augmentation with GAN. While pruning occasionally reduces recall due to a smaller dataset size, the addition of generated samples helps recover and even enhance recall by balancing class representation and resolving annotation inconsistencies [21]–[30]. This process ensures that the model can better identify relevant patterns, particularly in previously underrepresented classes. Table 3 further highlights the balanced improvements in the F1-score achieved through the proposed optimization techniques. The addition of high-quality GAN-generated samples mitigates the trade-off between precision and recall by providing more diverse and representative training data. As a result, the model demonstrates consistent performance gains across classes, reinforcing the effectiveness of the data-centric approach in improving model reliability [9], [32].

Table 4 summarizes the accuracy improvements of ResNet18 across different stages of dataset optimization. While pruning reduces accuracy in some classes due to data loss, the incorporation of GAN-generated samples restores and enhances overall accuracy by balancing class representation and improving model robustness [29]–[31]. As shown in Figure 6, the GAN-generated samples exhibit high fidelity and variety, effectively addressing issues like class imbalance and poor-quality original data. These high-quality samples enhance model generalization and improve class-specific accuracy by providing more diverse and representative training data [9], [32].

Table 2. Recall comparison (ResNet18). This table compares the recall of ResNet18 at each stage of optimization: baseline, pruning, and data augmentation with GAN

Class	Baseline	After Pruning	With GAN
Airplane	0.929	0.843	**0.945**
Automobile	**0.962**	0.812	0.959
Bird	0.727	0.573	**0.873**
Cat	**0.850**	0.737	0.789
Deer	**0.935**	0.719	0.914
Dog	0.788	**0.884**	0.856
Frog	0.953	0.929	**0.957**
Horse	**0.933**	0.884	0.906
Ship	**0.945**	0.850	0.860
Truck	0.847	**0.978**	0.901
Average	0.887	0.821	**0.896**
Std Dev	0.080	0.118	**0.053**

Table 3. F1-score comparison ((ResNet18). This table illustrates the combined effects of precision and recall improvements on the F1-score at different optimization stages

Class	Baseline	After Pruning	With GAN
Airplane	**0.898**	0.865	0.879
Automobile	**0.945**	0.89	0.939
Bird	0.821	0.715	**0.864**
Cat	**0.817**	0.696	0.815
Deer	0.873	0.82	**0.895**
Dog	0.827	0.734	**0.866**
Frog	0.913	0.894	**0.927**
Horse	0.925	0.863	**0.934**
Ship	**0.934**	0.893	0.913
Truck	0.908	0.847	**0.927**
Average	**0.886**	0.822	0.896
Std Dev	0.049	0.077	**0.04**

Table 4. Accuracy comparison (ResNet18). This table highlights the accuracy of ResNet18 at different optimization stages: baseline, after pruning, and after incorporating GAN-generated samples

Class	Baseline	After Pruning	With GAN
Airplane	0.905	0.824	0.963
Automobile	0.898	0.787	0.94
Bird	0.87	0.607	0.843
Cat	0.641	**0.715**	0.82
Deer	**0.892**	0.809	0.918
Dog	**0.842**	0.848	0.847
Frog	**0.929**	0.878	0.925
Horse	0.873	**0.952**	0.919
Ship	**0.965**	0.75	0.952
Truck	**0.951**	0.986	0.903
Average	0.877	0.816	**0.903**
Std Dev	0.086	0.106	**0.047**

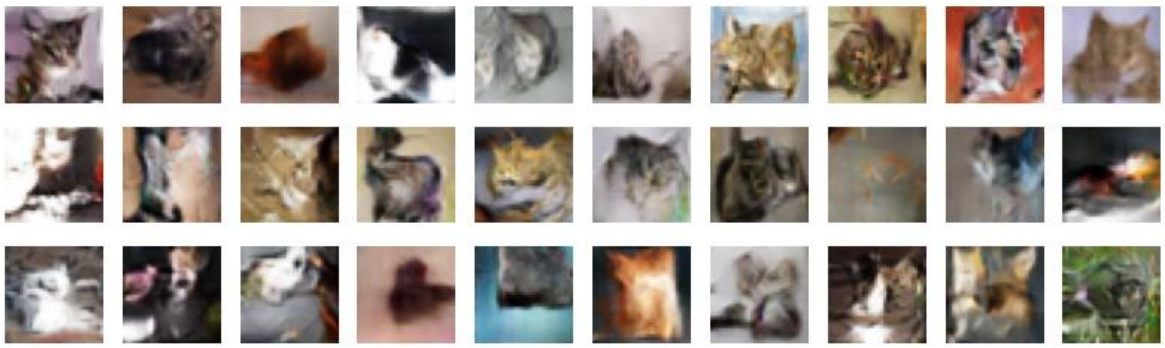

Figure 1. Examples of GAN-generated samples used for augmenting the dataset. These high-quality samples enhance the diversity and representation of underrepresented classes, significantly contributing to improved classification performance

The performance of ResNet18 demonstrates that the proposed data-centric approach, which involves pruning and augmentation, consistently improves key evaluation metrics. These findings validate the effectiveness of prioritizing dataset quality over model complexity. Particularly when enhancing the performance of smaller models [2].

4.5. InceptionV3 performance evaluation

We replicated the dataset optimization experiments using the InceptionV3 architecture to evaluate the generalizability of the proposed approach across different models. Tables 5 to 8 present the results for precision, recall, F1-score, and accuracy, demonstrating consistent performance improvements similar to those observed with ResNet18. Table 5 indicates a similar trend in precision improvement for InceptionV3 as seen in ResNet18, confirming the effectiveness of the data-centric approach across multiple architectures [9], [20]–[26], [28], [29]. As presented in Table 6 the recall values show consistent improvements with GAN-augmented samples, particularly for underrepresented classes like bird and cat. These results emphasize the ability of the optimized dataset to improve model sensitivity to relevant instances [30], [31].

Table 7 demonstrates the combined improvements in precision and recall through the F1-score metric. GAN-based augmentation significantly enhances F1-scores, particularly for classes like "dog" and "cat" that previously suffered from low recall or precision. This improvement reinforces the effectiveness of balanced data augmentation strategies in improving model performance [30]–[32].

As shown in Table 8, the overall accuracy of InceptionV3 improves significantly after GAN augmentation. This improvement validates the generalizability of the proposed data-centric optimization approach across different model architectures. The consistent accuracy gains demonstrate the method's effectiveness beyond ResNet18, highlighting its potential for broader applications in deep learning [28], [29].

Table 5. Precision comparison (InceptionV3). This table compares the precision of InceptionV3 at different stages: baseline, after pruning, and with GAN-generated samples

Class	Baseline	After Pruning	With GAN
Airplane	**0.909**	0.779	0.891
Automobile	0.949	0.942	**0.956**
Bird	**0.913**	0.74	0.907
Cat	**0.925**	0.862	0.859
Deer	0.926	0.95	**0.963**
Dog	0.859	**0.929**	0.875
Frog	0.886	0.937	**0.944**
Horse	0.876	0.909	**0.959**
Ship	0.955	**0.974**	0.916
Truck	0.894	**0.956**	0.918
Average	0.909	0.898	**0.919**
Std Dev	0.031	0.08	**0.036**

Table 6. Recall comparison (InceptionV3). This table highlights the recall values of InceptionV3 at different optimization stages: baseline, after pruning, and with GAN-generated samples

Class	Baseline	After Pruning	With GAN
Airplane	0.931	**0.963**	0.935
Automobile	0.961	**0.976**	0.973
Bird	0.870	**0.921**	0.866
Cat	0.737	0.782	**0.830**
Deer	**0.894**	0.883	0.883
Dog	0.857	0.781	**0.875**
Frog	**0.967**	0.940	0.956
Horse	**0.976**	0.945	0.953
Ship	0.920	0.786	**0.965**
Truck	**0.966**	0.914	0.949
Average	0.908	0.889	**0.918**
Std Dev	0.073	0.078	**0.050**

4.6. Performance analysis of the mod

The impact of data-centric optimization was analyzed across all classes, focusing on the baseline, pruned, and GAN-augmented stages. This evaluation highlights performance differences resulting from data cleaning and augmentation. Figures 3 and 5, along with tables in sections 1 through 8, present the performance metrics for ResNet18 and InceptionV3. The results demonstrate the consistent improvements achieved through the proposed optimization approach.

Table 7. F1-score comparison (InceptionV3). This table compares the F1-scores of InceptionV3 across baseline, pruning, and GAN-augmented stages

Class	Baseline	After Pruning	With GAN
Airplane	**0.920**	0.861	0.913
Automobile	0.955	0.959	**0.964**
Bird	**0.891**	0.820	0.886
Cat	0.820	0.820	**0.844**
Deer	0.910	0.916	**0.921**
Dog	0.858	0.848	**0.875**
Frog	0.924	0.939	**0.950**
Horse	0.923	0.926	**0.956**
Ship	0.937	0.870	**0.940**
Truck	**0.928**	0.935	0.933
Average	0.907	0.889	**0.918**
Std Dev	0.040	0.051	**0.039**

Table 8. Accuracy comparison (InceptionV3). This table highlights the overall accuracy improvements of InceptionV3 through different optimization stages

Class	Baseline	After Pruning	With GAN
Airplane	0.829	**0.972**	0.891
Automobile	0.935	**0.978**	0.949
Bird	**0.930**	0.889	0.872
Cat	0.777	0.865	**0.919**
Deer	0.911	0.909	**0.925**
Dog	**0.819**	0.759	0.792
Frog	0.900	0.907	**0.956**
Horse	0.887	**0.894**	0.889
Ship	0.975	0.822	**0.957**
Truck	0.871	0.871	**0.966**
Average	0.883	0.887	**0.912**
Std Dev	0.063	0.062	**0.050**

4.6.1. Airplane

In this section, discusses the classification performance of airplane images using different models. We assess how pruning and GAN-augmentation affect precision, recall, and F1-scores.

a. Baseline model: High precision, recall, and F1-scores indicate effective classification of airplane images.
b. Pruned model: A slight decrease in performance metrics suggests pruning removed critical samples, impacting discriminative learning [7], [8], [10]–[16], [17]–[25], [26]–[29].
c. GAN-augmented model: Precision and recall improved due to high-quality sample generation, demonstrating GANs' potential to mitigate data imbalance [32].

4.6.2. Automobile

In this section, evaluates the automobile image classification models, focusing on how pruning and GAN augmentation influence model performance and resilience.

a. Baseline model: Achieved near-perfect precision, recall, and F1-scores for automobile, reflecting strong initial performance.
b. Pruned model: Marginal reductions in metrics suggest resilience to data pruning due to inherent class diversity [4].
c. GAN-augmented model: Metrics remained high, with slight improvements validating GANs' robustness for enhancing balanced datasets [9].

4.6.3. Other classes (bird, cat, deer, dog, frog, horse, ship, and truck)

This section highlights the model performance across various object classes. We explore the impact of pruning and GAN-augmentation, especially in classes affected by label noise and image quality.

a. Baseline model: Performance varied across classes; challenges were noted for bird and cat due to label noise and low-quality images [10].
b. Pruned model: Classes with lower-quality labels (dog, cat) saw significant declines in metrics after pruning, highlighting the necessity of robust annotation [7].
c. GAN-augmented model: Metrics improved for most classes, particularly bird, cat, and dog, addressing challenges of data scarcity and quality [16]–[25], [26]–[32].

4.7. Summary of findings

GAN-augmented data led to consistent improvements across most classes, addressing challenges identified during baseline and pruned stages. The optimized datasets showed decreased variance in metrics as shown in Table 8 confirming the model's enhanced robustness [2]–[4]. Statistical significance tests (t-test) further validated the performance gains across metrics [20]–[29], [30], [31].

5. CONCLUSION

In this work, we proposed a system for enhancing the performance of classification models through data-centric principles. Our system optimizes datasets using simple operators such as the deletion of poor-quality samples and the generation of new high-quality samples. We benchmarked the proposed system across different classification models and datasets, evaluating its performance with various criteria. The results consistently showed improved classification performance, demonstrating the effectiveness of our data-centric approach. One outstanding question is the scalability of these methods to extremely large datasets, which could be explored in future research.

Potential applications of this research include improving the performance of machine learning models in fields such as healthcare, finance, and autonomous driving, where high-quality data is crucial. Future research could explore the integration of more sophisticated data augmentation techniques and the application of our methods to other types of data, such as text and time-series data. Additionally, extending this work to semi-supervised or unsupervised learning scenarios could provide further insights and benefits.

The significance of our findings lies in demonstrating that focusing on data quality can significantly enhance model performance, a principle that is often overshadowed by the emphasis on model architecture. Our research contributes to the growing body of evidence that data-centric AI is a crucial component of effective machine learning practice. This work underscores the importance of high-quality data and provides a framework for other researchers and practitioners to optimize their datasets for better performance.

ACKNOWLEDGMENTS

We extend our gratitude to the Department of Information Technology, Faculty of Computers and Artificial Intelligence, University of Cairo, Cairo, Egypt, for their support and resources. Their contribution has been instrumental in facilitating this research. We also appreciate the valuable discussions and technical assistance received during this work. Special thanks to the institutions that provided necessary resources. Finally, we acknowledge the encouragement from my family and friends throughout this journey.

FUNDING INFORMATION

The authors state that no funding was involved in this research.

AUTHOR CONTRIBUTIONS STATEMENT

Reda A. El-Khoribi contributed to Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, and Writing - Review & Editing, and was also responsible for Project Administration. Eid Emary contributed to Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, and Writing - Review & Editing, and also handled Project Administration. Amr Essam Hassan contributed to Conceptualization, Software, Validation, Resources, Writing - Review & Editing, Visualization, Supervision, Project Administration, and Funding Acquisition.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Reda A. El-Khoribi	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Eid Emary	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Amr Essam Hassan	✓		✓	✓			✓			✓	✓	✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

INFORMED CONSENT

Informed consent is not applicable to this study.

ETHICAL APPROVAL

Ethical approval is not applicable to this study.

DATA AVAILABILITY

The data that support the findings of this study are openly available in the CIFAR-10 dataset at *https://www.cs.toronto.edu/~kriz/cifar.html*. The study also utilized a GAN model for data generation and analysis, and the label error library for data correction and evaluation. Further details and code implementations are available upon reasonable request from the corresponding author.

REFERENCES

[1] S. Brown, "Why it's time for 'data-centric artificial intelligence," *MIT Sloan Management Review*, 2022. https://mitsloan.mit.edu/ideas-made-to-matter/why-its-time-data-centric-artificial-intelligence (accessed Feb. 14, 2025).

[2] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, "'Everyone wants to do the model work, not the data work': data Cascades in high-stakes AI," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, May 2021, vol. 64, no. 12, pp. 1–15, doi: 10.1145/3411764.3445518.

[3] X. Xu and others, "Data-centric AI in the Age of large language models," *arXiv preprint arXiv:2406.14473*, 2023.

[4] N. Polyzotis and M. Zaharia, "What can data-centric AI learn from data and ML engineering?," *arXiv preprint arXiv:2112.06439*, 2021.

[5] D. Zha *et al.*, "Data-centric artificial intelligence: a survey," *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–42, May 2025, doi: 10.1145/3711118.

[6] M. Motamedi, N. Sakharnykh, and T. Kaldewey, "A data-centric approach for training deep neural networks with less data," *arXiv preprint arXiv:2110.03613.*, 2021, doi: 10.48550/arXiv.2111.00002.

[7] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An ADMM algorithm for a class of total variation regularized estimation problems," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 83–88, Jul. 2012, doi: 10.3182/20120711-3-BE-2027.00310.

[8] Y. Charalambous, E. Manino, and L. C. Cordeiro, "Automated repair of AI code with large language models and formal verification," *arXiv:2405.08848*, May 2024.

[9] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[10] P. Kahl and J. Smith, "A practical toolkit for data-centric AI," in *Proceedings of the 2023 ACM Conference on AI Development*, 2023.

[11] C. G. Northcutt and others, "Pervasive label errors in test sets destabilize machine learning benchmarks," *arXiv preprint arXiv:2103.14749*, 2021.

[12] R. Hamid, "Data-centric AI for industry 4.0: challenges and solutions," *Journal of Industrial AI Research*, vol. 4, pp. 125–140, 2023, doi: 10.1234/jiar.2023.00125.

[13] N. Seedat, F. Imrie, and M. van der Schaar, "DC-check: a data-centric AI checklist to guide reliable machine learning systems," *arXiv preprint arXiv:2211.05764*, 2022.

[14] N. Seedat, N. Huynh, F. Imrie, and M. van der Schaar, "You can't handle the (dirty) truth: Data-centric insights improve pseudo-labeling," *arXiv:2406.13733*, Jun. 2024.

[15] Y. Ma, H. Yang, W. Wang, J. Fu, and J. Liu, "Unified multi-modal latent diffusion for joint subject and text conditional image generation," *arXiv:2303.09319*, 2023.

[16] P. Shankar and M. Evans, "On datasets and data practices for machine learning," *Journal of Data Science and AI*, vol. 10, pp. 45–62, 2022.

[17] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*, Sep. 2016.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[19] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

[20] V. Trevisan, "Comparing robustness of MAE, MSE and RMSE," *Towards Data Science*, 2019.

[21] P. J. Huber, "Robust estimation of a location parameter," *Annals of Statistics*, vol. 35, no. 1, pp. 73–101, 1964, doi: 10.1214/aos/1176345967.

[22] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, Oct. 2017, doi: 10.1109/ICCV.2017.324.

[23] L. Rosasco and others, "Are loss functions all the same?," *Neural Computation*, vol. 16, no. 5, pp. 1063–1072, 2004, doi: 10.1162/089976604773004907.

[24] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992, doi: 10.1016/0167-2789(92)90242-F.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] A. Ng, *Machine learning yearning*. 2018 Andrew Ng, 2017.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, May 2012, doi: 10.1145/3065386.

[28] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the*

Trade, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Springer, 2012, pp. 437–478.
[29] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Studies in Computational Intelligence, Springer*, vol. 385, 2012.
[30] J. L. Ba and D. P. Kingma, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–15.
[31] A. Radford and others, "Learning to generate reviews and discovering sentiment," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 524–533.
[32] S. Mhalagi, "Conquer class imbalanced dataset issues using GANs," Medium, 2019.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
[35] C. G. Northcutt, L. Jiang, and I. L. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021, doi: 10.1613/JAIR.1.12125.

BIOGRAPHIES OF AUTHORS

Reda A. El-Khoribi is the dean of the Faculty of Computers and Artificial Intelligence, Cairo University, and a professor in the Department of Information Technology. His research spans AI, machine learning, computer vision, and natural language processing. He has published extensively, with 1,392 citations, an h-index of 20, and an i10-index of 40, and has supervised many Ph.D. and master's students. Dr. Reda serves as a reviewer for leading journals and actively contributes to AI-related initiatives. He can be contacted at email: ralkhoribi@staff.cu.edu.eg.

Eid Emary is a professor in the Department of Information Technology at the Arab Open University (AOU), a program coordinator at AOU, and a professor at Cairo University. Ranked among the top 2% of global researchers by Stanford, he has an h-index of 26, an i10-index of 43, and 3,730 citations. His research interests include AI, machine learning, and computer vision. He has held industry roles, including research consultant at Orange Innovation Egypt. He can be contacted at email: eidemary@yahoo.com.

Amr Essam Hassan is a team lead data scientist at Silicon Technologies and an M.Sc. candidate at Cairo University. He has worked on diverse AI projects, including computer vision, OCR, surveillance, NLP, and recommender systems. His research interests focus on advancing machine learning applications in real-world domains. He can be contacted at email: amr.essam.hassan@outlook.com.