University of Münster

Department of Information Systems

Efficiency and Effectiveness of Deep Self-Learning With Multi-Prototypes From Noisy Labels

Seminar Thesis

in the context of the seminar

Advanced Topics in Machine Learning

submitted by

Leo Richard Hermann Giesen

Chair of Data Science:
Machine Learning and Data Engineering

Principal Supervisor	Prof. Dr. Fabian Gieseke
Supervisor	Jan Pauls, M.Sc.
	Chair for Data Science:
	Machine Learning and
	Data Engineering
Student Candidate	Leo Richard Hermann Giesen
Matriculation Number	462502
Field of Study	Information Systems
Contact Details	leo.giesen@uni-muenster.de
Submission Date	20.08.2024

Abstract

Large-scale datasets in real-world applications often suffer from noisy labels, which can significantly degrade the performance of deep learning models. This study investigates the efficiency and effectiveness of the deep self-learning approach suggested by [HLW19] that employs multiple representative class prototypes to iteratively correct noisy labels. By leveraging these prototypes to generate pseudo-labels, the Self-Learning with Multi-Prototypes (SMP) framework aims to robustly train networks on noisy datasets. However, this study's evaluation reveals that the proposed method, despite its potential, presents practical challenges. Training with SMP requires approximately double the time, and its complexity made it difficult to replicate the results reported in the original paper. Furthermore, the comparison with standard approaches is complicated by the fact that the traditional method overfitted the data, resulting in relatively poor test performance. These findings suggest that, while SMP may offer theoretical advantages, its practical application requires further refinement.

Contents

1 Motivation and Relevance

In machine learning, noisy labels are a frequent challenge in real-world applications. The collection of large-scale datasets with precise annotations is often expensive, time-consuming, and sometimes infeasible. Even human-labeled datasets can contain errors, leading to biases that degrade model performance. Consequently, large real-world datasets are typically noisy, negatively affecting the accuracy of models. To address this issue, several new approaches have emerged, with Self-Learning with Multi-Prototypes (SMP) showing particular promise for image classification [HLW19]. SMP stands out because it reduces noise by self-correcting labels. The method involves calculating multiple representative class data points, known as prototypes, to generate and refine corrected labels, called pseudo-labels, through iterative learning. SMP claims to be an efficient deep learning framework for training robust and accurate networks on large and noisy datasets.

To address the challenges posed by noisy labels, a solution must meet several critical requirements. It should exhibit dataset versatility and robustness, allowing it to be trained on various noisy datasets and different types of data. Additionally, it must demonstrate large-scale efficiency, effectively handling large datasets without compromising speed or resource usage. Most importantly, the solution should lead to improved performance, enhancing the accuracy and reliability of models trained on noisy data. However, testing the model on various datasets is beyond the scope of this study due to time and computing power constraints. Therefore, this research focuses on exploring the central question: How efficient and effective is deep self-learning with multi-prototypes when dealing with noisy labels? To address this question, it is essential first to define what is meant by efficiency and effectiveness within this context.

To thoroughly evaluate the SMP framework, we must clearly define what is meant by "efficiency" and "effectiveness" in the context of this study. Technical efficiency is defined as using the minimum necessary inputs to achieve a given output [EKD21; Cam24c; Far57, p. 6]. In this research, the critical inputs include resource usage, implementation time, inference time, and training duration (see Figure 1). Effectiveness refers to the ability to achieve the desired results [Cam24b; Cam24a; Dic24]. Specifically, in image classification, effectiveness is measured by the model's accuracy [Men23, Fig. 2]. Thus, within this study, the SMP approach will be considered effective if it demonstrates higher accuracy compared to standard approaches. These definitions will guide the subsequent analysis of the SMP framework's performance.

Efficiency		Effectiveness
Space Complexity	Resource Usage (Disk, GPU and System RAM Usage) Model Size	Performance Metrics (Accuracy, Precision, Recall, and F1-Score)
Time Complexity	Training Duration Inference Time Implementation Time	

Table 1 Efficiency and Effectiveness Evaluation Metrics of Machine Learning Approaches.

The relevance of this study goes beyond theoretical exploration, touching on significant practical and academic aspects. As AI becomes increasingly integrated across various industries, there is a growing demand for high-performance machine learning models [Sur24; Pre24; How23]. These models have substantial implications in the economic, environmental, social, consumer policy, and political domains [Inb24; BU21; Dom+23]. Achieving robust and accurate models, particularly when dealing with noisy labels, is crucial for ensuring their reliability and effectiveness in real-world applications.

Companies are investing heavily in the development of these high-performance AI models, reflecting the high expectations placed on their outcomes [Gol23; Sta+23]. This study is novel in its approach, being the first to examine the efficiency and effectiveness of the SMP framework in this context. This study's objective is to verify the claims made about this method, addressing the broader challenge of improving ML models' performance on noisy datasets, a critical issue faced in many practical scenarios. Ultimately, this research aims to contribute to the advancement of machine learning by enhancing the efficiency and effectiveness of models beyond current standards. By rigorously testing and evaluating the SMP approach, this study seeks to push the boundaries of what is achievable in the field of AI, offering insights that could influence future developments in both academic and industrial settings.

The subsequent sections are structured as follows: First, the theoretical background and related work are examined. Second, the SMP method and analysis approach are outlined. Third, the analysis results are presented and explained. Fourth, these findings are discussed and interpreted, with the limitations and implications of the results being considered. Finally, the findings are concluded and an outlook on future work is provided.

2 Related Work and Theoretical Background

The issue of noisy labels can be addressed through various approaches, which can be categorized into data augmentation and denoising, model-based techniques, loss function modifications, and label noise correction (see Table 2). *Data augmentation and denoising*, such as data augmentation and noise filtering and cleaning are typically time-consuming [SK19; ZW04; Joh24]. Thus, alternative and more viable approaches need to be considered in case of noisy and large datasets.

Centrality and Consistency (CC) is a *loss function modification* technique that enhances model robustness by enforcing consistent predictions in different augmented versions of the same input data [Zha+22]. It reduces the model's sensitivity to noise by penalizing discrepancies between predictions, thereby focusing the model on the most stable and relevant features [Zha+22]. While CC is particularly effective in semi-supervised learning and noisy environments, its success depends on the careful selection of augmentation strategies, as inappropriate augmentations can mislead the model.

Low-Rank Approximation with Consistency Constraints (LRA-diffusion CC) is a *model-based techniques* designed to enhance image classification by mitigating noise in large datasets through dimensionality reduction and iterative refinement [Che+23]. The approach leverages matrix decomposition to extract essential features while filtering out noise, followed by a diffusion process that smooths and refines these features, ensuring consistency throughout the dataset [Che+23]. Although it is the most effective approach on the given Clothing-1M dataset (see Figure 4), this method could be computationally intensive due to the iterative nature of the diffusion process, requiring careful tuning for optimal performance.

Training frameworks, such as SANM and the SMP approach, aim to maintain robustness in a large and noisy dataset. The SANM (Self-Adaptive Noise Modeling) approach within the DivideMix framework addresses noisy labels in large datasets by dynamically separating clean and noisy samples during training [Tu+23]. This method relies on a Gaussian Mixture Model (GMM) to estimate the probability that each sample is clean or noisy, treating noisy samples as unlabeled data within a semi-supervised learning context [Tu+23]. Although SANM is highly effective in managing label noise and enhancing model robustness, it can be computationally demanding and sensitive to the initial assumptions of the mixture model. The proposed SMP approach enhances the robustness of deep neural networks in noisy datasets by iteratively refining labels through a self-learning framework. This method uses multiple

prototypes to represent each class, allowing for better correction of mislabeled data without relying on assumptions about noise distribution or requiring additional clean data [HLW19].

Category	**Approach**	**Main Idea**
Data Augmentation and Denoising	Data Augmentation	Enhancing dataset size and variability through transformations
	Noise Filtering and Cleaning	Preprocessing data to identify and remove or correct noisy labels
Model-Based Techniques	LRA-Diffusion	Simplifies data and iteratively refines it through a diffusion process
Loss Function Modifications	Centrality and Consistency (CC)	Ensures consistent model predictions across data augmentations
Label Noise Correction	Self-Learning with Multi-Prototypes (SMP)	Iteratively corrects noisy labels through self-learning framework
	Self-Adaptive Noise Modeling (SANM)	Dynamically separates and refines noisy labels during training

Table 2 Selected Noise Handling Approaches for Image Classification.

The literature on noise-handling techniques in machine learning offers a variety of approaches, each with distinct strengths and challenges. However, several research gaps remain, particularly in the comprehensive evaluation of how these methods balance effectiveness and efficiency. One key area that my thesis aims to explore is the trade-off between model complexity and robustness to noisy labels, especially within model-based techniques where complex architectures are employed to manage noise. Although these methods are designed to reduce noise, they can inadvertently lead to overfitting, which increases sensitivity to noise and undermines the robustness of the model. In the context of the SMP approach, which iteratively refines labels to enhance their accuracy, a significant gap exists in understanding whether this refinement process truly improves model reliability without introducing inefficiencies or reinforcing incorrect labels. This research will specifically address the gap by evaluating how effectively and efficiently the SMP approach mitigates noise, providing insights that could guide future enhancements in noise-handling strategies.

3 Methods

3.1 Research Method

The methodology chapter details the approach used to rigorously evaluate the efficiency and effectiveness of deep self-learning with multi-prototypes from noisy labels, emphasizing the critical importance of reliable machine learning models in handling noisy datasets. Initial efforts to collaborate with the authors of the SMP approach [HLW19] were made to ensure accuracy and fidelity in the implementation. However, these attempts were unsuccessful. A comprehensive review of existing implementations was then conducted, revealing that only one implementation [Sar19] was available. This implementation did not utilize the Clothing-1M dataset used by the SMP approach and lacked the necessary results to address the specific research question of this study. In light of these limitations, the SMP approach was independently implemented to conduct a thorough evaluation (see Chapter A.5 and Repository [Gie24]). This independent implementation allowed direct control over the experimental parameters and ensured that the methodology aligned precisely with the research objectives. The objective is to train a two pretrained ResNet50 models: one using a standard training method and the other employing the SMP approach. Subsequently, the models efficiency and effectiveness evaluation metrics are compared to assess the efficiency and effectiveness of the SMP approach (see Table 1).

To replicate the results of the paper as accurately as possible, all of the following aspects were realized just like in the paper. All steps of the procedure are the same for both models with the exception of the training phase. The noisy dataset with 1,037,497 data points was acquired from Xiao (see Figure 6). Second, the data preparation comprised resizing the image with a short edge of 256 and randomly cropping it into 224 by 224 pixel image with 3 normalized RGB-channels because the images have different sizes. The dataset was generated from the file paths from the text files provided by the dataset. The tar files containing the images are extracted for fast image access (see example images in Figure 7). Han, Luo, and Wang did not mention their train-validation-test split. The data here was split into 65% training, 15% test and 20% validation data.[1] Both pretrained models are trained using an A100 GPU from Google Colab and use class balancing weights to handle class imbalance. Both models train for 15 epochs with a momentum of 0.9, a weight decay of 0.005, and a learning rate of 0.002, which decreases by 10 every 5 epochs (γ value of 0.1 and a step size of 5). The standard model uses a cross-entropy loss function, which is also

[1] The training set comprises 674,373 data points, while the validation set has 207,499 and the noisy test set has the remaining 155,625.

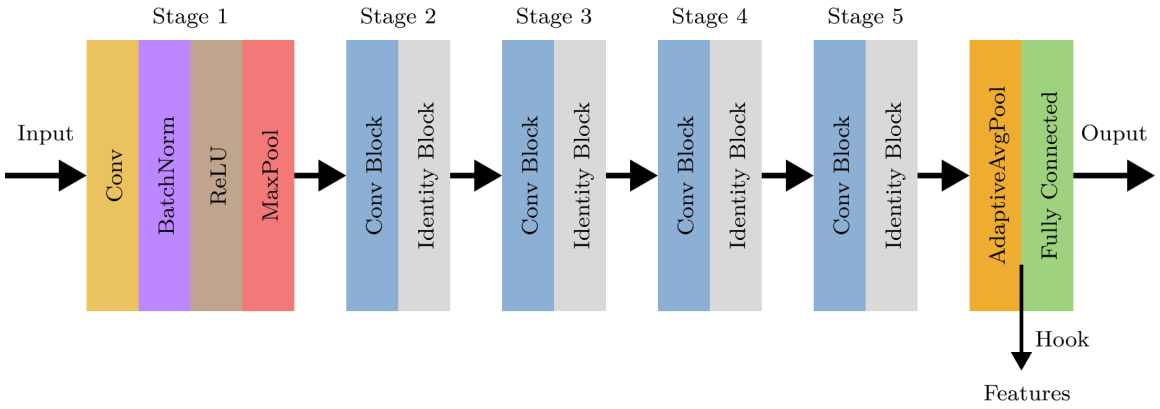


Figure 1 ResNet50 Model Architecture With Hook to Extract Features.

used by the SMP approach in the first iteration (see Equation A.1). In the subsequent iterations the SMP approach trains on both original and pseudo-labels equally. This is achieved by adapting the loss function to consider both labels and uses a weight factor α of 0.5 to balance the loss function for the original and corrected data (see Equation A.2). After each training epoch, the SMP approach corrects the labels using prototypes specific to each class. The explanation of how this is performed is already presented by Han, Luo, and Wang and is also summarized in Chapter A.2.

3.2 Method of Analysis

To evaluate the efficiency and effectiveness of deep self-learning with multi-prototypes from noisy labels, the statistic-based analysis uses the metrics visualized in Table 1. The efficiency metric resource usage refers to the utilization of hardware, for instance how many GB of RAM were used during training and inference. The image size, number of parameters and model size play a role in this space complexity metric. The time complexity efficiency metric training duration takes the epoch and total training duration including model latency (backward pass) and number of epochs to converge into consideration. Moreover, the inference time or model latency measures time it takes the model to predict with a forward pass of the model. Additionally, the time it takes to implement the approaches is compared, so that it becomes clear how simple and time-consuming it is to recreate the SMP approach on a different problem. Thus, the implementation efficiency and reusability of the approach is evaluated. The effectiveness of the SMP approach is evaluated against the standard approach by comparing their performances on the validation data using their accuracy, precision, recall and F1 score [Men23, Fig. 2]. It was beyond the scope to perform hyperparameter gridsearch to identify the optimal parameter values for α, the threshold percentile, the number of prototypes, and randomly sampled images.

4 Results

4.1 Effectiveness Evaluation

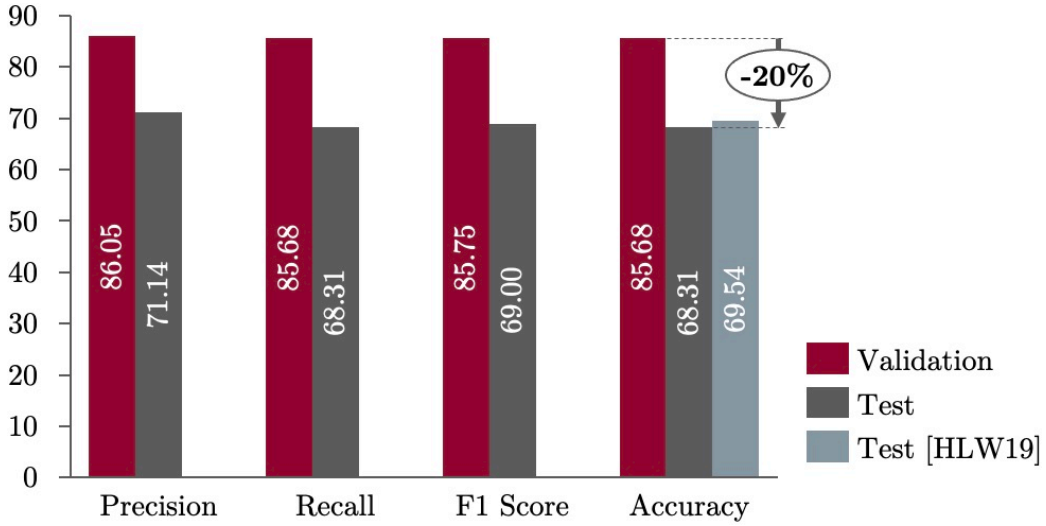


Figure 2 Effectiveness Evaluation of the Standard Approach in Percent.

This section presents the results from implementing and analyzing both the standard approach and the SMP approach, allowing for an evaluation of the SMP's effectiveness and efficiency. The models' effectiveness is assessed using accuracy as defined by Han, Luo, and Wang. The standard approach, utilizing cross-entropy loss, achieved an accuracy of 69.54% with the parameters specified in this study [HLW19, p. 5143].[2] The model trained in this study reached an accuracy of 68.31% on the clean test set and 85.84% on the noisy test set see Figure 2). The 1.23 percentage point difference from the clean test set can be attributed to uncertainties in the original paper, such as the lack of detailed information on image normalization, train-validation-test split, early stopping configuration, and the random seed used. These results suggest that the standard approach's outcomes were successfully replicated. Notably, the validation accuracy exceeded the test accuracy on clean data by approximately 25.43%. This significant drop between validation and test datasets was observed across all performance metrics. Furthermore, the clean test set accuracy of 68.31% is 24.66% lower than the training accuracy of 90.67%. Additionally, while the training loss continued to decrease, the validation loss began to plateau at epoch six (see Figure 9).

[2] Parameters for both approaches: 15 epochs, batch size of 128, momentum of 0.9, weight decay of 0.005, step size of 5, γ of 0.1 [HLW19, p. 5143].

The image classification task for the models on the Clothing-1M dataset involves categorizing images into 14 types of clothing.[3] The class-specific accuracy is presented in the confusion matrix (see Figure 8). The standard model struggled to distinguish between similar clothing items, such as knitwear and sweaters, or windbreakers, jackets, and downcoats. Additionally, the model often confused vests with dresses and downcoats with vests, though this confusion was not reciprocal. Among all classes, chiffon had the highest prediction accuracy, while jackets had the lowest with the standard approach.

Using the specified parameters,[4] Han, Luo, and Wang reported an accuracy of 74.37% with the SMP approach [HLW19, p. 5145]. In this study's SMP implementation, label correction resulted in approximately 91.40% of labels being altered, despite only 38.46% of the labels being noisy (see Figure 5b) [HLW19, p. 5143]. This suggests that the prototypes were neither representative nor diverse enough for their classes, leading to reduced accuracy and effectiveness. Due to the low quality of prototypes and the high GPU training costs incurred through Google Colab Pro+ (€116.34), SMP training was not completed, as there was no indication of potential performance improvement.

4.2 Efficiency Evaluation

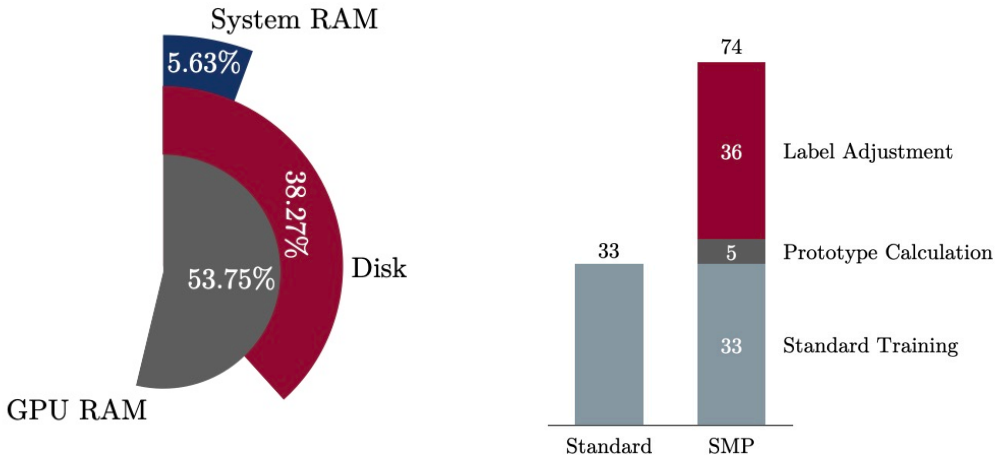
The efficiency of the models is evaluated based on space and time complexity. The model size, inference time, backward pass, and number of parameters (23,536,718) are identical for both models since the SMP approach only differs in its training procedure. Consequently, these variables have an identical impact on resource usage and training duration for both models. The resource usage includes disk, GPU, and system RAM consumption. The SMP approach utilized 4.7 GB of system RAM, 77 GB of disk space, and 21.50 GB of GPU RAM, resulting in utilization rates of 5.63%, 38.27%, and 53.75%, respectively.[5] Thus, it is evident that resource usage is not a limitation of the SMP approach and there are no significant hardware requirements or implications. Consequently, from a space complexity perspective, the SMP approach is viable.

From a time complexity perspective, the training duration and implementation time of both models were analyzed. The standard approach required approximately 33 minutes per epoch, while the SMP approach averaged 74 minutes per epoch (see

[3] Classes: t-shirt, shirt, knitwear, chiffon, sweater, hoodie, windbreaker, jacket, downcoat, suit, shawl, dress, vest, and underwear.

[4] SMP approach parameters: 320 randomly sampled images, eight prototypes per class, α of 0.5, 40^{th} threshold percentile.

[5] System RAM: 4.7/83.5 GB (5.63% utilized). Disk: 77.0/201.2 GB (38.27% utilized). GPU RAM: 21.5/40.0 GB (53.75% utilized).



(a) Resource Usage of the SMP Approach. (b) Epoch Training Duration in Minutes.

Figure 3 Efficiency Evaluation of Resource Usage and Epoch Duration.

Figure 3b). This doubling in epoch duration is primarily due to label adjustment and prototype calculation, with the prototype calculation accounting for only 12.20% of the increase, and label adjustment contributing the remaining 87.80%, largely due to the large size of the training set. The training converged at epoch 11, with the model training for about 4.40 hours and processing a total of 11,412,467 labels (see Figure 9). These findings indicate a high computational complexity and inefficiency in the SMP approach.

For both approaches, data preparation, model initialization, and evaluation were identical. The only difference lay in the training phase, where the SMP implementation was significantly more time-consuming than the standard approach. This inefficiency required extensive knowledge of the SMP approach, particularly in determining when to calculate specific values and accurately identifying the inputs and outputs of each function. Due to the complex interplay of variables and concepts, the implementation of the SMP approach required multiple days to complete.

	Efficiency	**Effectiveness**
Space Complexity	☻Resource Usage (Disk, GPU and System RAM Usage) ☻Model Size	☻Performance Metrics (Accuracy, Precision, Recall, and F1-Score)
Time Complexity	☻Training Duration ☻Inference Time ☻Implementation Time	

Table 3 Assessing Relative Efficiency and Effectiveness of SMP Approach with the Standard Approach (☻Lower, ☻Similar and ☻Higher Performance).

5 Discussion

5.1 Synthesis of Findings

This study investigates and evaluates the efficiency and effectiveness of deep self-learning with multi-prototypes (SMP) approach for handling noisy labels as proposed by Han, Luo, and Wang. In contrast to the standard approach, the *effectiveness* of the SMP approach from Han, Luo, and Wang could not be replicated in this study due to the low quality of the prototypes. Furthermore, the 17.53% discrepancy in test accuracy between clean and noisy datasets suggests that the model exhibits low noise tolerance, failing to demonstrate the intended robustness against noisy data (see Figure 4.1). Regarding *efficiency*, since the SMP approach only differs from the standard approach in its training methodology, their resource usage, model size, and inference time remain comparable. However, the SMP approach proved to be significantly inefficient in terms of training and implementation time, due to the high computational demands of label correction and the conceptual complexity involved. Furthermore, the SMP approach lacks easy reusability across various image classification tasks, as it requires extensive modifications to the code, increasing the risk of errors. These necessary adjustments involve changes to sampling strategies and hook positioning to fit different model architectures. Additionally, adapting to data complexity may require varying the number of prototypes to ensure accurate class representation, especially in more complex tasks.

The analysis of the SMP approach led to several suggestions for *improvement*. The training duration could be reduced by decreasing the number of randomly sampled images, as they significantly impact the time required for the prototype calculation. To further reduce training time, labels could be refined every five epochs rather than after each epoch. This adjustment could substantially cut down the training duration with only a minimal impact on accuracy, based on the assumption that predictions do not vary significantly between consecutive epochs.

To address the issue of similar classes being confused, it may be beneficial to introduce a confidence threshold for label changes. This could be implemented by comparing the average similarity score of the original class with the highest average similarity score of an alternative class. For example, if the latter score is at least 15% higher than that of the original class, the label would be changed; otherwise, the label would remain unchanged.

5.2 Implications, Limitations and Future Research Directions

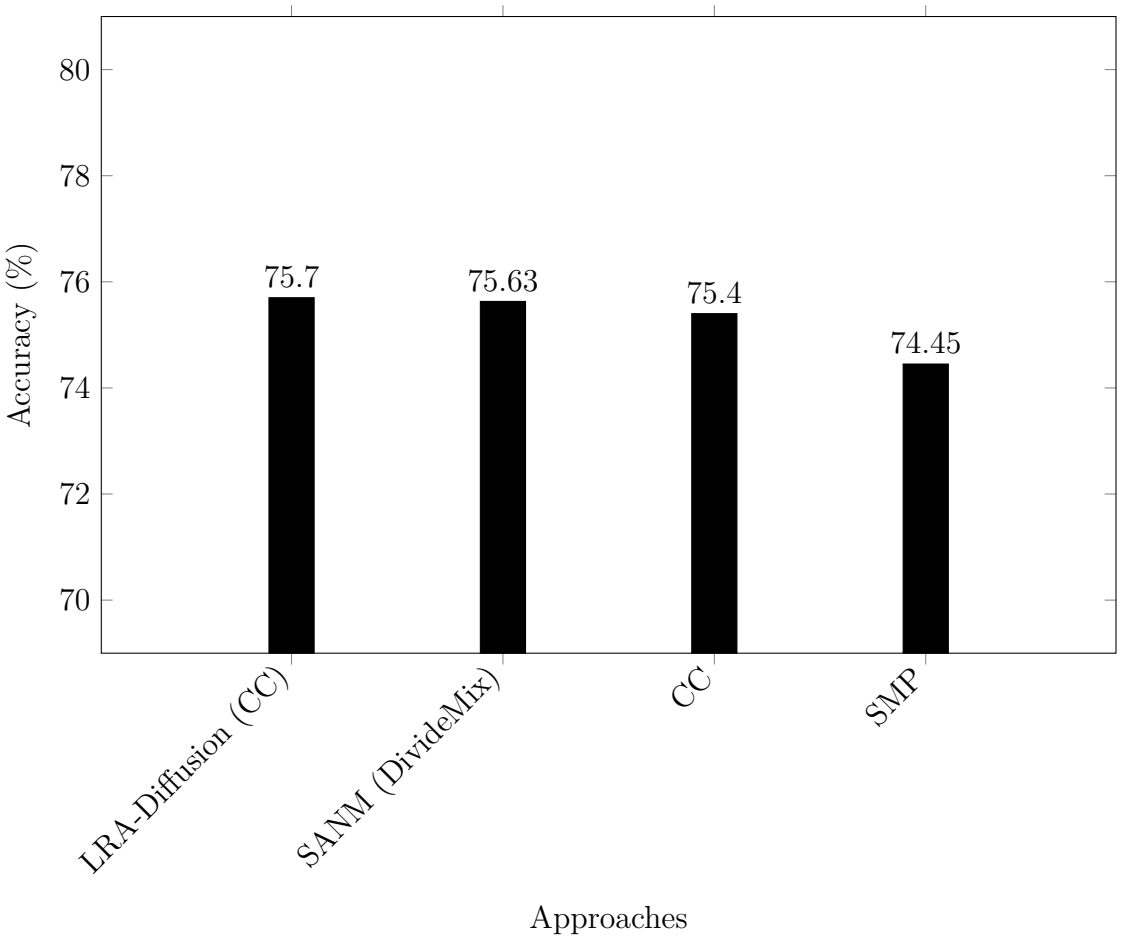


Figure 4 Accuracy of Noise Handling Approaches [Pap24].

As discussed in Chapter 2, there are numerous alternatives for managing noise in large datasets. At least 16 approaches, including the DivideMix method, achieve higher accuracy on the Clothing-1M test set than the SMP approach, which has an accuracy of 74.45% (see Figure 4) [HLW19; Pap24]. With accuracies reaching up to 75.7%, these methods demonstrate that the SMP approach is not state of the art. The inefficiency of the SMP approach likely contributes to its limited adoption in broader applications, which may explain its absence from the Papers With Code ranking.

The insights of this study are limited by four factors. First, the implementation time is influenced by the individual skills of the programmer and their prior knowledge of the data, making the evaluation of the implementation time subjective. Second, some statements from Han, Luo, and Wang could have been more carefully articulated to avoid potential misunderstandings. For instance, the assertion that "the concrete choice of S_c does not have influence in the final result because we only need the relative

density of images" [HLW19, p. 5141] may be misleading. The similarity threshold S_c is, in fact, significant. If set to 0 or 1, all images are treated as equally dense, which could lead to the selection of very similar prototypes reducing the coverage of class characteristics (see A.5). Additionally, Han, Luo, and Wang did not disclose all parameters of the SMP approach, such as the image normalization method, the train-validation-test split, the configuration of early stopping, or whether noisy or clean data was used for testing. Although many of these metrics could be inferred from commonly used values, there was insufficient transparency regarding the prototype calculation and label correction processes, which are critical to the effectiveness of the SMP approach. Attempts to contact Han, Luo, and Wang were unsuccessful, further limiting the study due to the semi-transparency of the SMP approach. Third, the occasional disconnection and reconnection of the GPU on Google Colab may have artificially increased the reported code execution times. This potential inaccuracy in time measurement cannot be entirely ruled out.

Fourth, the concern of overfitting in the standard model is analyzed. The steep drop in loss during training could be due to either a distribution shift or overfitting. A distribution shift would imply that the test set has different characteristics from the training and validation sets. However, the similarity in accuracy scores between this study's implementation and Han, Luo, and Wang's results makes a distribution shift unlikely, as it would require the data with the same characteristics to be randomly placed in the same datasets. Overfitting occurs when the model becomes overly tailored to the training data, learning noise patterns that do not generalize to new data [Yin19]. This is indicated by strong training accuracy but poor test accuracy, reflecting the model's inability to generalize effectively. Two observations suggest overfitting in the standard model with the parameters used by Han, Luo, and Wang [Ach24; MMC22]: a significant gap between training loss and validation loss, and the continued decrease in training loss while the validation loss plateaus (see Chapter 4.1). Thus, it is likely that the model overfitted, a concern that is not addressed by Han, Luo, and Wang. To ensure a critical analysis and fair comparison of the SMP and standard approaches, additional regularization is recommended, beyond the already applied early stopping and L2 regularization.

6 Conclusion and Future Work

6.1 Conclusion

This study evaluates the efficiency and effectiveness of the SMP approach in handling noisy labels. Despite showing efficiency in resource usage, model time, and inference time, the SMP approach displays critical inefficiencies. The training process is more than twice as long as the standard approach due to the repeated, time-consuming label adjustments. In addition, the code lacks reusability, as it requires multiple modifications depending on the dataset and the complexity of the image classification task. The implementation process also proved lengthy due to the conceptual complexity involved.

Furthermore, the study was unable to replicate the results from Han, Luo, and Wang due to the low quality of the prototypes generated by the SMP approach. Even if the SMP approach performs as claimed in the original paper, comparing it to the standard approach may be inequitable, as the standard model likely suffers from overfitting, which limits its generalization ability on the test dataset. This calls into question the significance of the original results, as better regularization could have improved the performance of the standard model. Moreover, the SMP approach is not as effective as several other noise-handling methods available for large, real-world datasets [Pap24].

In conclusion, the SMP approach analyzed in this study exhibits significant limitations that undermine its efficiency and effectiveness in large, noisy datasets. The lack of substantial performance improvements does not justify the extended training duration and implementation efforts required. Therefore, alternative methods that better address noise handling should be considered.

In pursuit of advancing the state-of-the-art in image classification with noisy labels, this study has revealed significant limitations and shortcomings of the SMP approach. These findings provide valuable insights that can guide future research directions in the development of more efficient and effective noise-handling methods. However, the insights from this study are constrained by several factors, including the subjectivity of the implementation time metric, the semi-transparency of [HLW19], potential areas for implementation improvement, limitations inherent to Google Colab, and the possibility of an inequitable comparison of the models' effectiveness.

6.2 Future Work

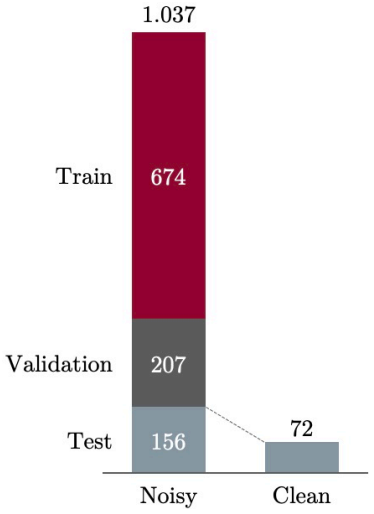
To enhance the SMP approach, several improvements could be considered. First, higher quality prototypes may be achieved by incorporating the smallest distance to a single prototype, rather than relying solely on the average similarity score. Additionally, while 320 randomly sampled images appear sufficient, it would be beneficial to test the minimum number of images required to maintain high accuracy, thereby reducing computational complexity and saving time. Starting from the second epoch, prototype quality could be further improved by sampling only images where the pseudo label matches the original label, ensuring that noisy labels are excluded from prototype calculations.

Further research into the SMP approach could also focus on developing a guide to determine the optimal number of prototypes, as data complexity increases. Although Han, Luo, and Wang have already provided guidance for the Clothing-1M dataset, the optimal number of prototypes may change with more complex datasets. Additionally, the SMP approach may face scalability challenges, as training duration tends to increase exponentially with model complexity, given that all data is processed in each epoch. Other areas for future research include monitoring prototype consistency during training to ensure prototype quality and stability. Furthermore, an analysis of label noise mitigation could be conducted by examining the proportion of noisy labels that are correctly versus incorrectly adjusted by the model.

As the field of noise-handling techniques continues to evolve, future research should focus on refining prototype selection methods and improving scalability in the SMP approach. By doing so, the gap between current methodologies and practical, real-world applications can be bridged, ensuring more reliable and efficient image classification in noisy environments.

A Appendix

A.1 Methodology



(a) Train-Test-Validation Data Split (in Thousands).

(b) Proportion of Noise in the Clothing-1M Dataset.

Figure 5 Quantitative Overview of the Dataset.

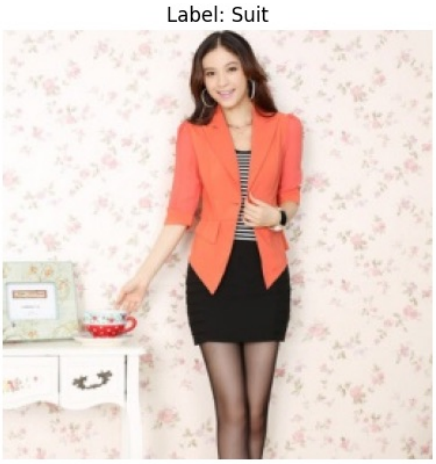


Figure 7 Data Samples of the Clothing-1M Dataset.

A.2 SMP Approach

The number of prototypes selected is determined by the complexity of the image classification problem, with more complex problems requiring a greater number of prototypes to accurately represent class features. Because this study works with the same dataset of the SMP approach, the same parameters, such as the number of

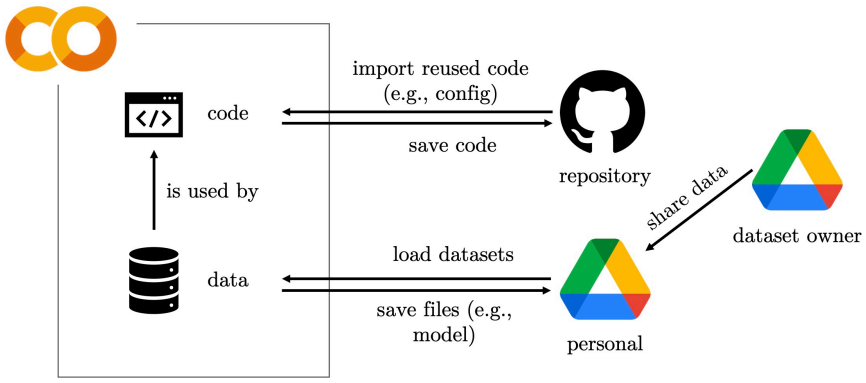


Figure 6 Architectural Setup: Integration of Services.

prototypes (eight prototypes) have been selected [HLW19, p. 5143]. The prototypes for each class are calculated by analyzing the typical features associated with each class. The intermediate output just before the fully connected layer is treated as the features of the input image's class [HLW19, p. 5141]. To accurately represent a class, many images from each class were randomly selected. Han, Luo, and Wang experimented with sampling 320, 640, 1280 and 2560 images per class, which resulted in similar accuracy performance scores [HLW19, p. 5145]. To minimize computational costs, 320 images per class were sampled and processed individually through the model. The features of each image were then extracted using a hook (see Figure 1).

Next, the cosine similarity of the features is calculated and compiled into a similarity matrix. This matrix is used to calculate the feature densities ρ, which are determined by counting the number of images with a cosine similarity above the threshold value S_c and subtracting those below it. The threshold value S_c is set at the 40^{th} percentile of the similarity values in the matrix [HLW19, p. 5142]. The similarity matrix and densities are used to identify representative and diverse prototypes using the similarity measurement η.

A high density data point is likely to be representative of its class because many data points within the same class share similar features. However, if nearly identical data points are selected exclusively, the full range of class features may not be represented. To ensure that a prototype with the highest density $\rho_i = \rho_{\max}$ covers a wide range of class characteristics, the lowest similarity measurement $\min_j S_{ij}$ serves as an accurate indicator of the prototype's suitability. This approach verifies that the selected prototype represents *diverse* features by ensuring that it is moderately distinct. If the data point density is not the maximum ($\rho_i < \rho_{\max}$), the highest similarity between i and a denser data point j is used. This metric evaluates the data point's usefulness as a prototype. This metric indicates how well i is connected to a denser and more reliable feature space, thereby determining if the data sample is a *representative* and reliable prototype (see Equation A.3). After calculating the similarity measurement for each

image, the highest scoring prototypes are selected. The label of the class prototypes' features with the highest average similarity σ_c to the image's features is selected as the image label (see Equation A.7). After the training, both models are evaluated on the noisy test set and a clean test set to understand how well the approaches work on noisy and clean data, which was provided in the Clothing-1M dataset (see Figure 5a).[6] The evaluation encompasses a performance analysis, employing accuracy, precision, recall, and the F1-score.

A.3 SMP Approach Calculations

$$\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), y) = -\frac{1}{n} \sum_{i=1}^{n} \log\left(\mathcal{F}\left(\theta, \mathbf{x}_i\right)_{y_i}\right) \tag{A.1}$$

A.1 Standard Cross-Entropy Loss Function with the Prediction $\mathcal{F}(\theta, \mathbf{X})$ and Original Noisy Labels y [HLW19, p. 5140].

$$\mathcal{L}_{\text{total}} = (1-\alpha)\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), y) + \alpha\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}), \hat{y}) \tag{A.2}$$

A.2 SMP Cross-Entropy Loss Function with the Pseudo-Labels $\hat{y}$ and the Weight Factor α [HLW19, p. 5140].

$$\eta_i = \begin{cases} \max_{j, \rho_j > \rho_i} S_{ij}, & \rho_i < \rho_{\max} \\ \min_j S_{ij}, & \rho_i = \rho_{\max} \end{cases} \tag{A.3}$$

A.3 Similarity Measurement (Prototype Usefulness Metric) With Density ρ and Similarity S of the Data Points i and j [HLW19, p. 5142]

$$S_{ij} = \frac{\mathcal{G}\left(\mathbf{x}_i\right)^T \mathcal{G}\left(\mathbf{x}_j\right)}{\left\|\mathcal{G}\left(\mathbf{x}_i\right)\right\|_2 \left\|\mathcal{G}\left(\mathbf{x}_j\right)\right\|_2} \tag{A.4}$$

A.4 Cosine Similarity with Features $\mathcal{G}(\mathbf{x})$ (Used for Equation A.5) [HLW19, p. 5142]

$$\rho_i = \sum_{j=1}^{m} \operatorname{sign}\left(S_{ij} - S_c\right) \tag{A.5}$$

A.5 Density (Used for Equation A.3) [HLW19, p. 5142]

[6] The clean test set comprises 72,409 data points compared to the 155,625 data points of the noisy dataset.

$$\operatorname{sign}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0. \end{cases} \tag{A.6}$$

A.6 Sign Calculation (Used for Equation A.5) [HLW19, p. 5142]

$$\sigma_c = \frac{1}{p} \sum_{l=1}^{p} \cos\left(\mathcal{G}(\mathbf{x}), \mathcal{G}\left(\mathbf{x}_{cl}\right)\right), c = 1 \ldots K \tag{A.7}$$

A.7 Average Cosine Similarity of Image's Features $\mathcal{G}(\mathbf{x})$ and Class Prototypes' Features $\mathcal{G}(\mathbf{x}_{cl})$ with the c-th Class and p-th Prototype [HLW19, p. 5142].

A.4 Results

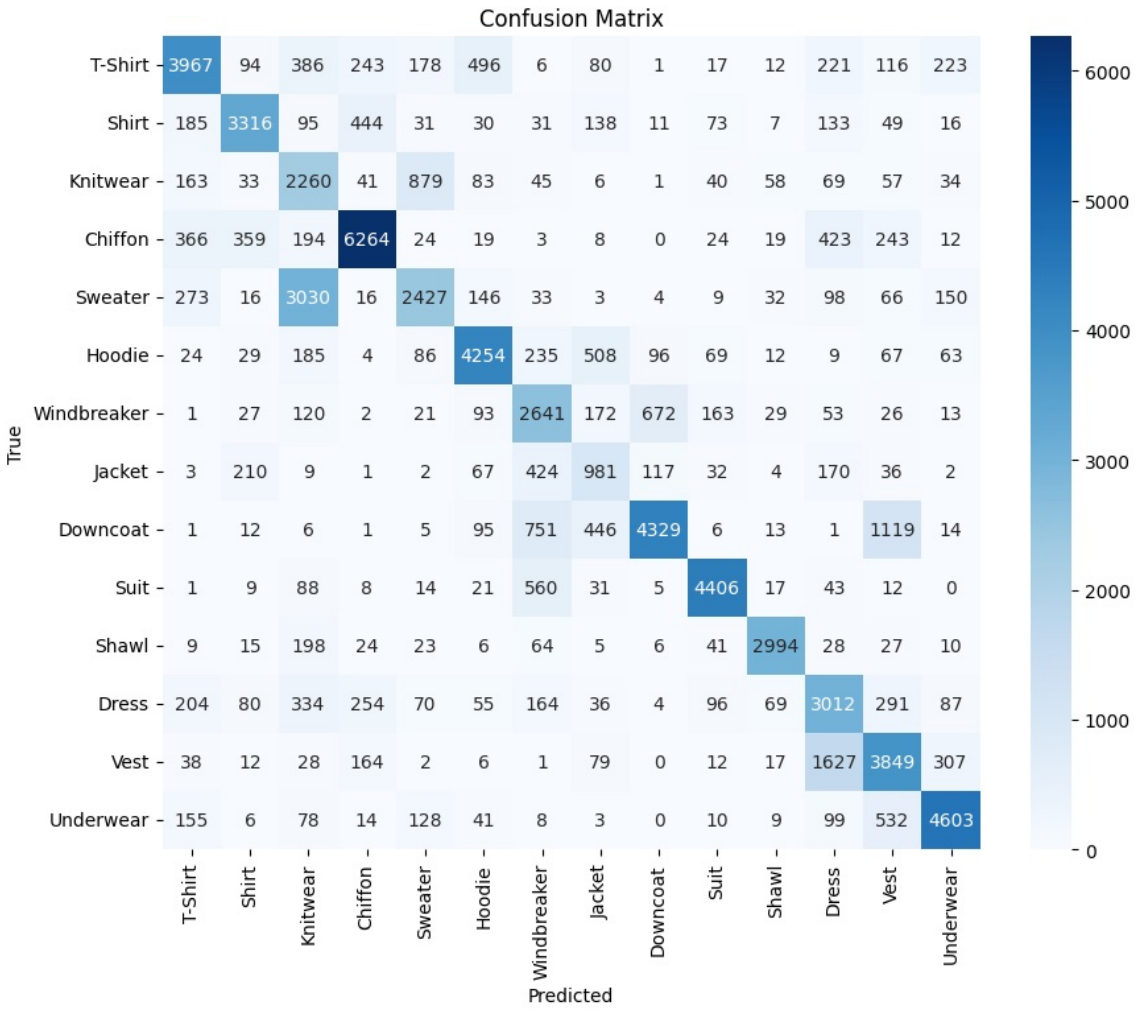


Figure 8 Confusion Matrix of the Standard Approach.

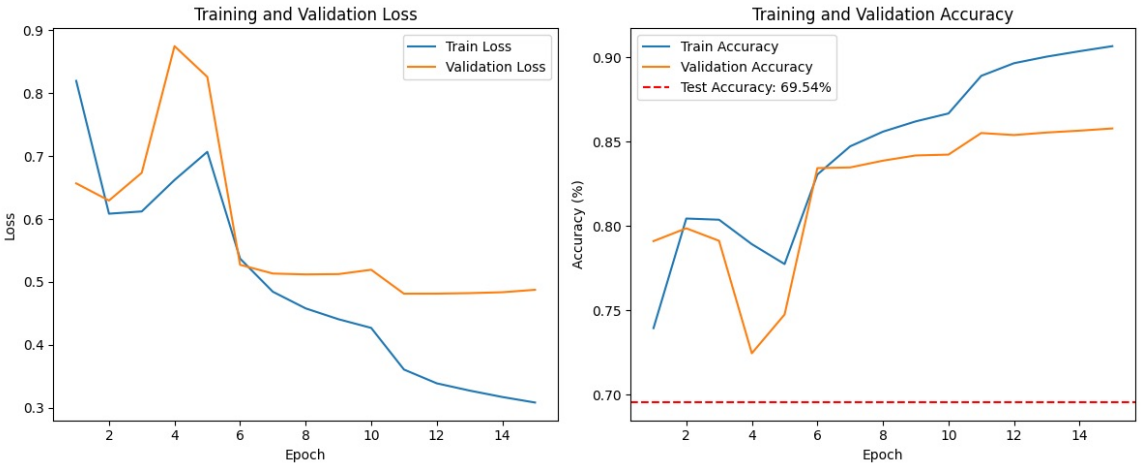


Figure 9 Training Loss and Accuracy Convergence of the Standard Approach on the Noisy Dataset.

A.5 Code

Algorithm 1 ResNet50 Model Hook.

```
# define the hook function with the
# intermediate_output as the features
def hook(module, input, output):
    global intermediate_output
    intermediate_output = output.detach()

def reset_intermediate_output():
    global intermediate_output
    intermediate_output = None

# register the hook to the layer
# before the FC layer (AdaptiveAvgPool2d)
hook = model.avgpool.register_forward_hook(hook)
```

Algorithm 2 SMP Label Correction Phase Code (Continued in Algorithm 3).

```
# 1. train the model
for epoch in range(num_epochs):
    model.train() # set model to training mode
    for inputs, labels in train_loader:
        # move input and label tensors to the device
        inputs = inputs.to(device)
        labels = labels.to(device)
        # zero out the optimizer
        optimizer.zero_grad()
        # forward pass
        outputs = model(inputs)
        # loss is computed differently in first epoch
        if epoch == 0:
            loss = criterion(outputs, labels)
        else:
            # 2.2 label correction
            pseudo_labels = []
            for inputs, labels in train_loader:
                # reset intermediate_output
                reset_intermediate_output()
                model(inputs)
                # extract features of the samples with the hook
                # (automatically saved as intermediate_output)
                batch_pseudo_labels = correct_labels(prototypes,
                    intermediate_output, labels)
                pseudo_labels.extend(batch_pseudo_labels)
            # calculate the loss on pseudo and original labels
            loss_original = (1-alpha) * criterion(outputs, labels)
            loss_pseudo = alpha * criterion(outputs, pseudo_labels)
            loss = loss_original + loss_pseudo
```

Algorithm 3 SMP Prototype Calculation Code.

```
# 2. label correction phase (except in last iteration)
with torch.no_grad():
    if epoch != num_epochs-1:
        # 2.1 prototype selection
        # initialize prototypes
        prototypes = []
        for class_id in range(num_classes):
            # Reset the features
            reset_intermediate_output()
            # sample images for the current class
            sample_loader = sample_images(class_id)
            # Feed the randomly sampled images through the model
            # to automatically extract features into the
            # intermediate_output variable using the hook
            samples_features = []
            for inputs in sample_loader:
                inputs = inputs.to(device)
                model(inputs)
                samples_features.append(intermediate_output)
                reset_intermediate_output()
            # convert to tensor for following operations
            samples_features = torch.cat(samples_features)
            # calculate similarity matrix of sample features
            similarity_matrix = cos_similarity(
                samples_features, randomly_sampled_img_count)
            # calculate the threshold
            threshold = calc_similarity_threshold(
                similarity_matrix, threshold_percentile)
            # calculate density of images
            densities = calc_rho_densities(
                similarity_matrix, threshold)
            # calculate similarity measurement
            eta = calc_eta_similarity_measurement(
                similarity_matrix, densities)
            # select prototypes for each class
            class_prototypes = select_prototypes(
                eta, samples_features, num_prototypes)
            prototypes.append(class_prototypes)
        prototypes = torch.cat(prototypes)
scheduler.step()
```

Bibliography

[Ach24] Akruti Acharya. *Overfitting in Machine Learning Explained*. 2024. URL: https://encord.com/blog/overfitting-in-machine-learning/ (visited on 08/19/2024).

[BU21] Pol Borrellas and Irene Unceta. "The Challenges of Machine Learning and Their Economic Implications". In: *Entropy* 23.3 (2021), p. 275.

[Cam24a] Cambridge Dictionary. *Effective | English Meaning*. 2024. URL: https://dictionary.cambridge.org/dictionary/english/effective (visited on 08/16/2024).

[Cam24b] Cambridge Dictionary. *Effectiveness | English Meaning*. 2024. URL: https://dictionary.cambridge.org/dictionary/english/effectiveness (visited on 08/16/2024).

[Cam24c] Cambridge Dictionary. *Efficiency | English Meaning*. 2024. URL: https://dictionary.cambridge.org/dictionary/english/efficiency (visited on 08/16/2024).

[Che+23] Jian Chen et al. "Label-Retrieval-Augmented Diffusion Models for Learning from Noisy Labels". In: *Advances in Neural Information Processing Systems* 36 (2023).

[Dic24] Dictionary.com. *Effectiveness | Definition and Meaning*. 2024. URL: https://www.dictionary.com/browse/effectiveness (visited on 08/16/2024).

[Dom+23] Andrés Domínguez Hernández et al. "Ethical, Political and Epistemic Implications of Machine Learning (Mis)Information Classification: Insights From an Interdisciplinary Collaboration Between Social and Data Scientists". In: *Journal of Responsible Innovation* 10.1 (2023).

[EKD21] Obsa Teferi Erena, Mesfin Mala Kalko, and Sara Adugna Debele. "Technical Efficiency, Technological Progress and Productivity Growth of Large and Medium Manufacturing Industries in Ethiopia: A Data Envelopment Analysis". In: *Cogent Economics & Finance* 9.1 (2021), p. 1997160.

[Far57] Michael James Farrell. "The Measurement of Productive Efficiency". In: *Journal of the Royal Statistical Society: Series A* 120.3 (1957), pp. 253–281.

[Gie24] Leo Giesen. *Deep Self-Learning From Noisy Labels*. 2024. URL: https://github.com/lgiesen/Deep-Self-Learning-From-Noisy-Labels (visited on 08/17/2024).

[Gol23] Goldman Sachs. *Worldwide: AI Investment Growth 2025*. 2023. URL: https://www.statista.com/statistics/1424667/ai-investment-growth-worldwide/ (visited on 08/16/2024).

[HLW19] Jiangfan Han, Ping Luo, and Xiaogang Wang. "Deep Self-Learning From Noisy Labels". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 5138–5147.

[How23] Howard. *Integration of High-Performance Computing and Artificial Intelligence*. 2023. URL: https://community.fs.com/article/integration-of-highperformance-computing-and-artificial-intelligence.html (visited on 08/16/2024).

[Inb24] InbuiltData. *The Evolution and Impact of Machine Learning Models in Industry*. 2024. URL: https://www.linkedin.com/pulse/evolution-impact-machine-learning-models-industry-inbuiltdata-0mxoc/ (visited on 08/16/2024).

[Joh24] Ritu John. *What Is Data Augmentation: A Comprehensive Guide*. 2024. URL: https://www.docsumo.com/blogs/data-extraction/data-augmentation (visited on 08/20/2024).

[Men23] Gaurav Menghani. "Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better". In: *ACM Computing Surveys* 55.12 (2023), pp. 1–37.

[MMC22] Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. "Overfitting, Model Tuning, and Evaluation of Prediction Performance". In: *Multivariate Statistical Machine Learning Methods for Genomic Prediction* (2022), pp. 109–139. DOI: 10.1007/978-3-030-89010-0_4.

[Pap24] Papers With Code. *Clothing1M Benchmark (Image Classification)*. 2024. URL: https://paperswithcode.com/sota/image-classification-on-clothing1m (visited on 08/19/2024).

[Pre24] J.P. Pressley. *HPC Processing Boosts AI Innovations Across Industries*. 2024. URL: https://biztechmagazine.com/article/2024/07/hpc-processing-boosts-ai-innovations-across-industries (visited on 08/16/2024).

[Sar19] Sarsbug. *SMP: Pytorch implementation for Deep Self-Learning From Noisy Labels*. 2019. URL: https://github.com/sarsbug/SMP (visited on 06/10/2024).

[SK19] Connor Shorten and Taghi M Khoshgoftaar. “A Survey on Image Data Augmentation for Deep Learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48.

[Sta+23] Stanford University et al. *Total Global AI Investment 2015-2022*. 2023. URL: https://www.statista.com/statistics/941137/ai-investment-and-funding-worldwide/ (visited on 08/16/2024).

[Sur24] Surf. *High-Performance Machine Learning: Efficient and Scalable Machine Learning in HPC Environments*. 2024. URL: https://www.surf.nl/en/high-performance-machine-learning-efficient-and-scalable-machine-learning-in-hpc-environments (visited on 08/16/2024).

[Tu+23] Yuanpeng Tu et al. “Learning With Noisy Labels via Self-Supervised Adversarial Noisy Masking”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2023), pp. 16186–16195.

[Xia16] Tong Xiao. *Noisy Label: Code for the CVPR15 Paper "Learning From Massive Noisy Labeled Data for Image Classification"*. 2016. URL: https://github.com/Cysu/noisy_label/tree/master (visited on 06/10/2024).

[Yin19] Xue Ying. “An Overview of Overfitting and Its Solutions”. In: *Journal of physics: Conference series*. Vol. 1168. IOP Publishing. 2019, p. 22022.

[Zha+22] Ganlong Zhao et al. “Centrality and Consistency: Two-Stage Clean Samples Identification for Learning with Instance-Dependent Noisy Labels”. In: *Lecture Notes in Computer Science* (2022), pp. 21–37.

[ZW04] Xingquan Zhu and Xindong Wu. “Class Noise vs. Attribute Noise: A Quantitative Study”. In: *Artificial intelligence review* 22 (2004), pp. 177–210.

Declaration of Authorship

I hereby declare that, to the best of my knowledge and belief, this thesis titled *Efficiency and Effectiveness of Deep Self-Learning With Multi-Prototypes From Noisy Labels* is my own, independent work. I confirm that each significant contribution to and quotation in this thesis that originates from the work or works of others is indicated by proper use of citation and references; this also holds for tables and graphical works.

Münster, 20.08.2024

Leo Richard Hermann Giesen

Consent Form

Name: Leo Richard Hermann Giesen
Title of Thesis: Efficiency and Effectiveness of Deep Self-Learning With Multi-Prototypes From Noisy Labels

What is plagiarism? Plagiarism is defined as submitting someone else's work or ideas as your own without a complete indication of the source. It is hereby irrelevant whether the work of others is copied word by word without acknowledgment of the source, text structures (e.g. line of argumentation or outline) are borrowed or texts are translated from a foreign language.

Use of plagiarism detection software. The examination office uses plagiarism software to check each submitted bachelor and master thesis for plagiarism. For that purpose the thesis is electronically forwarded to a software service provider where the software checks for potential matches between the submitted work and work from other sources. For future comparisons with other theses, your thesis will be permanently stored in a database. Only the School of Business and Economics of the University of Münster is allowed to access your stored thesis. The student agrees that his or her thesis may be stored and reproduced only for the purpose of plagiarism assessment. The first examiner of the thesis will be advised on the outcome of the plagiarism assessment.

Sanctions Each case of plagiarism constitutes an attempt to deceive in terms of the examination regulations and will lead to the thesis being graded as "failed". This will be communicated to the examination office where your case will be documented. In the event of a serious case of deception the examinee can be generally excluded from any further examination. This can lead to the exmatriculation of the student. Even after completion of the examination procedure and graduation from university, plagiarism can result in a withdrawal of the awarded academic degree.

I confirm that I have read and understood the information in this document. I agree to the outlined procedure for plagiarism assessment and potential sanctioning.

Münster, 20.08.2024

Leo Richard Hermann Giesen