Clustering-based Coreset Selection for Dataset Distillation

Abstract

Dataset distillation, a technique that condenses large datasets into smaller, representative subsets while preserving model performance, has recently gained attention in deep learning. The DREAM (Dataset distillation by REpresentAtive Matching) Liu et al. (2023) approach, which relies on K-Means clustering for sample selection and matching, has shown impressive results. However, K-Means Lloyd (1982) clustering has limitations, such as handling noisy data and instability. This report explores alternative clustering methods to enhance DREAM's performance. We investigate various techniques, including K-Means++ Arthur and Vassilvitskii (2007), K-medoids Kaufman and Rousseeuw (1987), DBSCAN Ester (1996), and Agglomerative Johnson (1967), both with and without weighted samples, to find more robust solutions for dataset distillation. These methods offer insights into the dataset distribution, grouping similar data points, highlighting variations, and reducing complexity. Our primary goal is to determine if alternative clustering methods can better represent the distribution of the entire dataset and further enhance the dataset distillation process. The experiments reveal that among all clustering methods tested, K-Means++ performs exceptionally well, even without weighted samples, consistently outperforming the baseline, and weighted K-Means also outperforms the baseline. These findings underscore the potential of alternative clustering methods to enhance dataset distillation and model performance.

1 Introduction

In recent years, dataset distillation has emerged as a compelling technique in the field of deep learning. This approach involves condensing vast datasets into smaller, yet representative subsets, all while preserving the performance of machine learning models. Dataset distillation aims to synthesize small datasets with minimal information loss from original large-scale ones, ultimately leading to reduced storage and training costs.

One notable achievement in this realm is the DREAM (**Dataset distillation by REpresentAtive Matching**) method, which has demonstrated remarkable results. DREAM leverages K-Means clustering as a pivotal element of its strategy, employing REpresentAtive Matching to synthesize these condensed datasets.

DREAM's methodology involves selecting representative samples from the original dataset using K-Means clustering. These selected samples are then matched during the distillation process to create a compact yet informative dataset. This innovative approach has shown great promise in effectively reducing dataset sizes while maintaining model performance.

However, as with any methodology, K-Means clustering possesses its own set of limitations. Firstly, it struggles to effectively handle noisy data and is susceptible to instability. Secondly, the selection of a representative subset could potentially benefit from the inclusion of weighted samples, an aspect that K-Means clustering does not inherently address. Recent research has demonstrated that augmenting the selection process with weighted samples can yield improved outcomes.

Preprint. Under review.

Given the significance of weighted samples in data distillation, our research explores the performance of alternative clustering methods both with and without weights. Specifically, we investigate **K-Means++**, **K-medoids**, **DBSCAN**, and **Agglomerative** without weights, while also examining **K-Means**, **K-Means++**, **K-medoids**, **DBSCAN**, and **Agglomerative** with weights.

In light of these considerations and the default usage of K-Means without weights in DREAM, our aim is to explore the potential advantages and disadvantages of weighted and unweighted clustering methods. We intend to assess how these approaches impact dataset distillation performance and whether they can provide more robust and adaptable solutions.

Clustering Estivill-Castro (2002), as an exploratory data analysis technique, holds significant promise in providing insights into the overall distribution of the original dataset. It offers several advantages, including the grouping of similar data points to reveal underlying patterns, highlighting variations and commonalities within the data, and ultimately reducing the complexity of the dataset.

By undertaking a comprehensive comparative analysis of these clustering methods, our objective is to enhance the efficiency and effectiveness of dataset distillation techniques. We seek to determine whether alternative clustering methods, both with and without weights, can better represent the distribution of the entire dataset, thus contributing to the advancement of deep learning research. In essence, we aspire to shed light on the potential of alternative clustering approaches in improving the dataset distillation process and its ability to accurately capture the dataset's underlying distribution.

2 Related Work

2.1 K-Means

Despite its popularity, K-Means Lloyd (1982) has drawbacks. It is sensitive to noise and outliers, can distort cluster shapes, requires user input for the initial number of clusters, and is unsuitable for categorical data. The K-Means++ variant addresses some issues by improving the selection of starting clusters, but users should be cautious about replicated results across iterations.

2.2 K-Medoids

Unlike K-Means, which calculates the mean to determine the cluster center, k-medoids Kaufman and Rousseeuw (1987) computes the cluster center using an actual point. This distinction is significant as it avoids potential issues associated with extreme values. The primary objective of the k-medoids algorithm is to minimize the average dissimilarity of objects against their closest object.

It is noteworthy that k-medoids utilizes a greedy search strategy, which may result in failure to find an optimum solution. Despite this limitation, the k-medoids method is advantageous for its resilience against outliers and its ability to provide a more robust representation of cluster centers.

2.3 DBSCAN

DBSCAN Ester (1996)stands out as one of the most popular density-based algorithms. Unlike centroid-based methods, DBSCAN requires that the density in the neighborhood of an object should be sufficiently high to designate it as part of a cluster. The algorithm creates a cluster skeleton with a set of core objects possessing overlapping neighborhoods. Points inside the neighborhood of core objects delineate the boundary of clusters, while the remaining points are treated as noise. The algorithm necessitates two parameters: 1) ϵ, which denotes the starting point, and 2) MinPts, the minimum number of points required to form a dense region.

While DBSCAN performs well against noise and handles clusters of varied shapes, it may face challenges in high-dimensional datasets and is sensitive to the MinPts parameter. Despite these, it stands as a favorable alternative to K-Means in certain scenarios.

2.4 Agglomerative

Cluster analysis in this category involves generating a sequence of nested partitions, visualized as a tree or hierarchy known as a cluster dendrogram. The hierarchical tree exhibits different levels

of abstraction, with leaves at the lowest level and the root at the highest. Each leaf node represents a cluster for individual points, while the root encompasses all points in a single cluster. The dendrogram can be cut at intermediate levels to obtain meaningful clusters.

The hierarchical clustering approach falls into two classes: agglomerative Johnson (1967) and divisive. Agglomerative algorithms, often used, merge the most similar pair of clusters in a bottom-up manner. Various agglomerative algorithms differ in how they update the conflict of similarity between existing and merged clusters, with methods such as single linkage and complete linkage being well-known.

2.5 Dataset Distillation

Dataset distillation refers to the process of selecting a subset of data samples that encapsulate the most vital and representative characteristics of the original dataset. By distilling the dataset, we aim to create a smaller, refined version that retains the statistical properties and generalization capabilities of its larger counterpart.

3 Methods

Our project introduces an innovative dataset distillation method by incorporating DREAM (Dataset distillation by REpresentAtive Matching) (Liu et al., 2023) and leveraging a suite of clustering algorithms to enhance the model training process.

Guided by the DREAM framework, our dataset distillation process focuses on constructing a synthesized dataset S to effectively represent the entire original dataset. We initialize dataset S by clustering each class into 'N' clusters, adhering to a predefined image per class (ipc). The initial synthetic images are selected as either the centroids or the data points closest to the centroids of these clusters.

In contrast to the original DREAM algorithm, which utilizes only K-Means clustering, our approach introduces diversity by applying various clustering methods within each class to establish 'K' clusters, corresponding to the batch size. During each synthesis iteration, the selected points within these clusters undergo a crucial step of gradient matching. This iterative process refines the synthetic images, ensuring they faithfully capture the learning signal of the original dataset. The goal is to adjust the synthetic images so that the gradient updates during the training process closely resemble those of the full dataset, enhancing the representativeness of the distilled dataset.

3.1 Dataset distillation with gradient matching

Given an extensive dataset $T = \{(x_i^t, y_i^t)\}_{i=1}^{|T|}$, the goal of dataset distillation is to craft a compact surrogate dataset $S = \{(x_i^s, y_i^s)\}_{i=1}^{|S|}$ with minimal information loss, where $|S| \ll |T|$. The metric for information loss is reflected in the performance decline observed when training a model on the original images T compared to the surrogate set S.

Typically, optimization-based methods follow a synthetic pipeline. Initially, S is seeded with randomly chosen original images from T. Adhering to constraints set by matching objectives $\phi(\cdot)$, synthetic images undergo adjustments to mimic the distribution of the original images. This process is framed as:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathbf{D}\left(\phi(\mathcal{S}), \phi(\mathcal{T})\right) \tag{1}$$

Here, $\mathbf{D}$ represents the matching metric, often incorporating training gradients as the objective $\phi(\cdot)$. Drawing inspiration from the approach presented in (**?**), we introduce gradient matching. For a randomly initialized model M_θ with training parameters θ, S is expected to yield gradients akin to T throughout the training of M_θ, expressed as:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathbf{D}\left(\nabla_\theta \mathcal{L}(\mathcal{M}_\theta(\mathcal{A}(\mathcal{S}))), \nabla_\theta \mathcal{L}(\mathcal{M}_\theta(\mathcal{A}(\mathcal{T})))\right) \tag{2}$$

Here, $\mathcal{L}(\cdot, \cdot)$ denotes the training loss, and $\mathcal{A}$ signifies the differentiable augmentation. In practice, matching objectives are computed on synthetic images and a mini-batch of original images sampled

from T with identical class labels. The iterative alternation of objective matching and M_θ training forms the inner optimization loop, involving different randomly initialized M_θ for a diverse range of matching gradients. The term *Weight* is introduced, indicating the size of the centroid corresponding to a cluster. This iterative process with gradient matching enhances the representativeness of the distilled dataset S.

4 Experiment

4.1 Experimental Settings

In order to compare the results with baseline model and optimize dataset distillation methods, we conducted several experiments using the CIFAR-10 dataset, which encompasses 10 diverse classes, each comprising 5,000 training images and 1,000 testing images.

The chosen of neural network for our study was ConvNet-3, which is 3-layer convolutional networks. The clustering methods we pick were K-Means, serving as the baseline, along with K-Means++, DBSCAN, K-Medoids, and Agglomerative clustering. Each of these algorithms was tested for its ability to enhance the training process, both with and without the integration of weight adjustments.

The experimental procedure involved running 1,000 iterations for each clustering method, with each iteration comprising 100 inner loops dedicated to the gradient matching process. This amounted to an extensive 100,000 gradient matching operations in total. To assess the influence of clustering over the training period, clustering operations were conducted at every tenth inner loop.

4.2 Performance Comparison

We compared the performance of different clustering methods in the dimensions of accuracy and loss. Four groups of experiment were conducted: 1/10 images per class and Weight=True/False. Each result was compared to the baseline with the same amount of images per class, where the baseline method used K-Means clustering without weights. Data collected from the experiments is as follows:

Clustering Method	Weight	Accuracy	Loss
K-Means	True	50. 19%	22. 42
K-Means (Baseline)	False	50. 09%	22. 20
K-Means++	True	49. 51%	20. 49
K-Means++	False	49. 43%	20. 45
K-Medoids	True	44. 31%	17. 33
K-Medoids	False	44. 00%	17. 17
DBSCAN	True	45. 07%	47. 96
DBSCAN	False	46. 00%	41. 83
Agglomerative	False	42. 36%	18. 67

Table 1: Results of 1 image per class, compared with baseline

Clustering Method	Weight	Accuracy	Loss
K-Means	True	66. 76%	4. 71
K-Means (Baseline)	False	66. 51%	4. 78
K-Means++	True	66. 87%	4. 25
K-Means++	False	66. 91%	4. 23
K-Medoids	True	57. 47%	4. 49
K-Medoids	False	57. 67%	4. 50
DBSCAN	True	52. 73%	18. 50
DBSCAN	False	52. 82%	17. 81
Agglomerative	False	56. 86%	4. 87

Table 2: Results of 10 images per class, compared with baseline

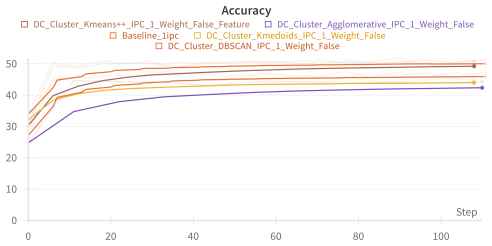

Figure 1: Accuracy of Weight = False, 1ipc.

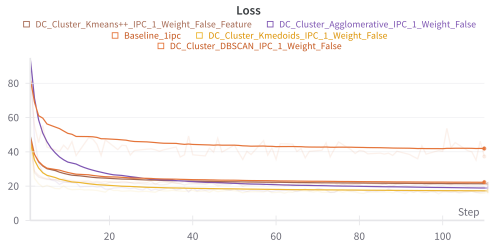

Figure 2: Loss of Weight = False, 1ipc.

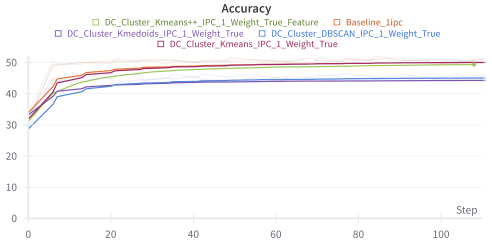

Figure 3: Accuracy of Weight = True, 1ipc.

Figure 4: Loss of Weight = True, 1ipc.

For training groups with 1 image per class and no weights, the baseline method yielded the best accuracy of 50. 09%. Known for its improved initialization over standard K-Means, K-Means++ performed comparably to the baseline with a similar accuracy of 49. 43%. However, other clustering methods such as K-Medoids, DBSCAN, and Agglomerative performed much worse than the baseline. With only one image per class, these clustering methods may require more information to perform effectively, as they rely on more complex distance measures or hierarchical structures.

Losses of all clustering methods, except DBSCAN, were generally lower than that of the baseline. Since accuracies did not improve over the baseline, this decrease in losses was likely due to overfitting.

When training with weights, we expected the accuracies to rise if the clustering methods can utilize the additional information provided by weights. These weights could help algorithms focus on the data points that are more representative to the entire dataset , thereby improving the generalizability of the distilled dataset. However, in the results of 1 pic we have not seen any significant improvement on accuracies: K-Means was improved from 50. 09% to 50. 19%, while K-Means++ was improved from 49. 43% to 49. 51%.

Comparing the loss graphs for weights set to true and false respectively, we concluded that the most significant difference lay in the stability and final convergence values. The training group with weights showed a smoother convergence process and relatively lower loss values than the group without weights. This suggested that even though the addition of weights did not significantly change the overall pattern of loss reduction, it may contribute to a more stable convergence, potentially indicating a more robust model fitting.

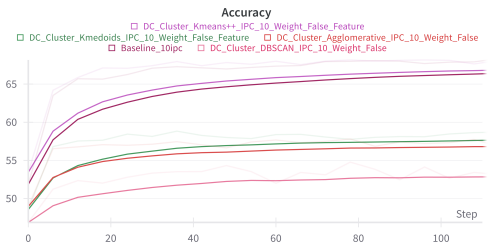

Figure 5: Accuracy of Weight = False, 10ipc.

Figure 6: Loss of Weight = False, 10ipc.

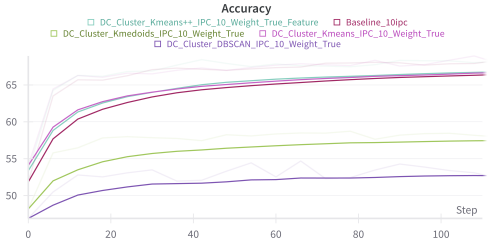

Figure 7: Accuracy of Weight = True, 10ipc.

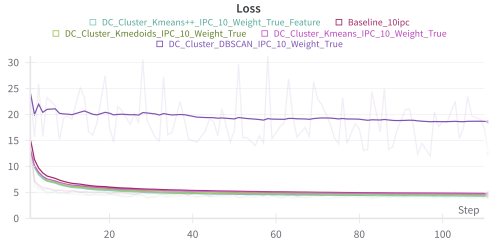

Figure 8: Loss of Weight = True, 10ipc.

For training groups with 10 image per class and no weights, the accuracy of the baseline is around 66. 51%. Among all clustering methods, K-Means++ significantly outperformed the baseline with an accuracy at 66. 91%. The increase in data volume from 1 ipc to 10 ipc provided a richer and more diverse representation of each class and allowed for better learning and generalization. With more data points in each class, K-Means++ could leverage its initialization advantage more effectively, leading to better clustering results from the outset.

Losses of all clustering methods, except DBSCAN, were relatively similar to the baseline. K-Medoids decreased the loss of baseline by 0.2 and K-Means++ decreased by 0.5, indicating that the robustness of training was improved.

Unlike the performance of trainings with 1 ipc, 10 ipc groups demonstrated more prominent improvements after assigning weights. Weighted K-Means and K-Means++ both outperformed the unweighted K-Means baseline with accuracies of 66. 76% and 66. 87%, respectively. The performance boost aligned with our expectation that extra information provided by weights could help prioritize representative samples and enhance their influence. Also, such weights could mitigate noise and outliers, leading to more robust models.

With weights, losses of K-Means, K-Means++, and K-Medoids all dropped below the baseline. The weighted K-Means++ had the lowest training loss of 4.24, roughly 0.5 lower than the baseline.

In conclusion, models trained with 10 images per class generally performed better than those trained with 1 image per class. The positive impact of weight assignment is more significant in 10 ipc experiments than 1 ipc ones. Among all clustering methods tested, weighted K-Means and K-Means++ were more likely to outperform the baseline. On the other hand, the performances of DBSCAN in four groups were consistently unsatisfactory. This is possibly because density-based clustering is not suitable to high-dimensional image data. The sizes of clusters generated by the Agglomerative method may be largely biased, indicating that it may not be an optimal choice for data distillation tasks. For K-Medoids at 1 and 10 ipc, training losses during the matching process were lower than the baseline accuracies were not improved possibly due to overfitting.

5 Discussion

5.1 Weighted Sampling

Weighted sampling demonstrated remarkable performance improvements. The inclusion of weighted samples enhances the representation of individual data points within clusters, making

them more informative during the dataset distillation process. Weighted sampling allows for a finer-grained selection of data points, preserving essential information while reducing dataset size.

5.2 K-Means++

K-Means initialization methods significantly impact clustering performance. K-Means++ strategically choose centroids to improve convergence speed and clustering quality. Thoughtful initialization helps reduce noise and enhance the efficiency of the clustering process.

5.3 K-Medoids

K-Medoids algorithm, known for its robustness in handling diverse datasets, exhibits a tendency towards overfitting, prompting the need for careful parameter tuning. However, one drawback is its relatively slower computational speed, notably hindering acceleration with GPU technologies.

5.4 DBSCAN

DBSCAN faces challenges with inconsistent batch sizes, limited capacity to learn subtle patterns, and struggles to accurately represent complex data distributions. These limitations can impact its performance, especially with datasets of varying sizes and intricate structures.

5.5 Agglomerative

Agglomerative clustering may exhibit an initial bias towards compact or dense structures, particularly when employing linkage criteria that favor the merging of close or similar entities. This bias can impact the resulting cluster formation, potentially leading to the preferential grouping of nearby data points.

5.6 K-Means++ Success

K-Means++ emerged as the top-performing clustering method in our experiments. This success can be attributed to its non-random selection of initial cluster centroids, which results in a more representative subset of data. By carefully initializing cluster centers, K-Means++ excels in capturing the underlying dataset distribution.

5.7 Challenges with K-Medoids

K-medoids, on the other hand, exhibited suboptimal performance in our experiments. Several factors contribute to this outcome. Firstly, K-medoids may lead to overfitting, as it can transform moderately confident model predictions into less confident ones. This overfitting effect can undermine the reliability of model outputs. Secondly, K-medoids selects only one point per cluster as its center, which might not fully capture the cluster's characteristics. Finally, our experimentation suggests that K-medoids may be better suited for gradient space clustering rather than the feature space we utilized.

Future research should delve deeper into the potential advantages of gradient space clustering, paving the way for more robust dataset distillation techniques in the field of deep learning.

5.8 Limitations

The poor performance of DBSCAN may because it does not represent the entire distribution of the dataset, and it cannot determine the batch size or virtual center points. K-Medoid shows poor results and tends to overfitting. K-Means++ initialization points are not random but more representative. Assigning weights to points makes a point more representative of the entire cluster's information.

6 Conclusion

At 1 ipc, employing Kmeans++ with and without weights, along with the original K-Means with weights, results in comparable validation accuracy to the baseline and yields a lower training loss of

approximately 1. All clustering methods except DBSCAN had low losses. Without any significant accuracy boost, such decrease in losses was likely due to overfitting instead of performance improvement. These clustering methods might need additional information to function efficiently with only one image per class, as they depend on complex distance measures or hierarchical structures. Comparing the losses of training groups with and without weights, we can also conclude that adding weights may contribute to a more stable convergence and likely more robust fitting.

With more images in class, 10 ipc training groups generally performed better than 1 ipc ones. Unlike models trained with 1 ipc, those trained with 10 ipc demonstrated more significant improvements by weight assignment, aligning with our purpose for adding weights to data points. At 10 ipc, incorporating weights into the original K-Means or utilizing K-Means++ (with or without weights) can enhance validation accuracy by approximately 0. 5% and reduces training loss by about 0. 5.

In the four training groups (1/10ipc, with/without weights), alternative clustering methods such as DBSCAN, Agglomerative, and K-Medoids all fail to achieve comparable or superior testing accuracy when compared to K-Means. The potential reasons for the unsatisfactory results are as follows:

- DBSCAN: Density-based clustering may be unsuitable for the distillation of high-dimensional image data as the selected subsets fail to accurately represent the entire dataset. High-dimensional data often leads to sparsity, which makes it difficult for density-based algorithms to identify meaningful clusters, as these algorithms depend on finding regions of high density.
- Agglomerative: The generated cluster sizes may exhibit significant bias, rendering them inadequate in representing the entirety of the dataset. The curse of dimensionality may also negatively affect the performance of the agglomerative clustering method.
- K-Medoids: During the matching process at 1 and 10 ipc, the training loss is lower than the baseline, but the accuracy is substantially inferior, possibly indicating overfitting issues. K-Medoids may be more suitable for the gradient space rather than the feature space.

7 Future Works

Although our study has tested the performance of K-Means, K-Means++, K-Medoids, DBSCAN, Agglomerative methods with and without weigts, there are other clustering methods available to explore, such as BIRCH. With a broader scope of analysis, further studies can draw a more comprehensive conclusion on the differences of performances among different clustering methods. Also, if computability permits, future studies can consider evaluating the clustering performance under different conditions, specifically exploring scenarios with 50 ipc and 100 ipc. According to the different results between 1 ipc and 10 ipc, we infer that with more data points per class, there may be notable improvements or variations in clustering outcomes.

Given that it is difficult to train K-Medoids clustering extensively, developing a torch version of K-Medoids can accelerate training and enhance the efficiency of the algorithm by leveraging the power of GPU processing. Trained with more data, K-Medoids may exhibit a comparable performance as K-Means and K-Means++. In addition, only applying clustering methods on the feature space, we expect the training results of gradient space to differ due to potential insights into the final condensing results. Applying clustering methods on the gradient space can potentially uncover patterns overlooked by feature space analysis.

References

Lloyd, S. (1982) Least squares quantization in PCM. *IEEE Transactions on Information Theory*, **28**(2), 129-137. `https://ieeexplore.ieee.org/document/1056489`

Kaufman, L., & Rousseeuw, P. J. (1987) Clustering by means of medoids. *Statistical data analysis based on the L1-norm and related methods*, **37**, 405-416. `https://www.sciencedirect.com/science/article/pii/S0167947387800324`

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96 Proceedings*, 226-231.

Johnson, S. C. (1967) Hierarchical clustering schemes. *Psychometrika*, **32**(3), 241-254. https://doi.org/10.1007/BF02289588

Pisinger, D. (1999) A Faster Exact Algorithm for Knapsack with Partition Constraints. *Operations Research*, **47**(3), 497-503. DOI: 10.1287/opre.47.3.497

Liu, Y., Gu, J., Wang, K., Zhu, Z., Jiang, W., & You, Y. (2023) DREAM: Efficient Dataset Distillation by Representative Matching. *arXiv preprint arXiv:2302.14416*, https://arxiv.org/abs/2302.14416.

Arthur, D., & Vassilvitskii, S. (2007). K-Means++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035.

Estivill-Castro, Vladimir (20 June 2002). "Why so many clustering algorithms – A Position Paper". *ACM SIGKDD Explorations Newsletter*, 4(1), 65–75. doi:10.1145/568574.568575. S2CID 7329935.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen (2021) *Dataset Condensation with Gradient Matching*. *arXiv:2006.05929 [cs.CV]*.