Augment & Valuate : A Data Enhancement Pipeline for Data-Centric AI

Youngjune Lee[1], **Oh Joon Kwon**[2], **Haeju Lee**[2],
Joonyoung Kim[3], **Kangwook Lee**[3], **Kee-Eung Kim**[1, 2]
[1]School of Computing, KAIST, Daejeon, Republic of Korea
[2]Kim Jaechul Graduate School of AI, KAIST, Daejeon, Republic of Korea
[3]Samsung Research, Republic of Korea
dudwns511@kaist.ac.kr, {ojkwon, hjlee}@ai.kaist.ac.kr,
{joon0.kim, kw.brian.lee}@samsung.com, kekim@kaist.ac.kr

Abstract

Data scarcity and noise are important issues in industrial applications of machine learning. However, it is often challenging to devise a scalable and generalized approach to address the fundamental distributional and semantic properties of dataset with black box models. For this reason, data-centric approaches are crucial for the automation of machine learning operation pipeline. In order to serve as the basis for this automation, we suggest a domain-agnostic pipeline for refining the quality of data in image classification problems. This pipeline contains data valuation, cleansing, and augmentation. With an appropriate combination of these methods, we could achieve 84.711% test accuracy (ranked #6, Honorable Mention in the Most Innovative) in the Data-Centric AI competition only with the provided dataset.

1 Introduction

Data scarcity and noise are important issues in industrial applications of machine learning. These issues have been discussed a lot in the past and there are various studies. To deal with scarcity of data, methods such as auto-augmentation (Cubuk et al. [2019], Lim et al. [2019], Hataya et al. [2020]), GAN-based augmentation (Antoniou et al. [2017]), few-shot learning (Finn et al. [2017], Chen et al. [2019]) and various other methods are being developed.

Another direction of research aims to improve data quality by focusing on learning objective or model architecture such as training with noisy data (Patrini et al. [2017], Lee et al. [2019]) and biased data (Saito et al. [2020], Ovaisi et al. [2020], Rosenbaum and Rubin [1983]). Moreover, there are practical methods that focus on data quality for data engineering, such as reinforcement learning based (Yoon et al. [2020]) and influence function based data valuation (Chen et al. [2021], Koh and Liang [2017]). Still, researches on data do not gain much attention compared to the evolution of the model architectures and algorithms.

Data valuation, augmentation and representation learning (Chen et al. [2020], Chen and He [2021]) are known to train a robust model. We consider this from a different point of view and intend to use these approaches to construct a pipeline that can improve the data quality for data-centric AI. We propose this pipeline, and achieved 84.711% test accuracy (ranked #6 and won an Honorable Mention) in Data-Centric AI competition only with the provided training data and an appropriate combination of previously mentioned methods.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

(a) (b) (c) (d)

Figure 1: Example of noisy data. (a), (b) are mislabeled.
(a) GT(Ground Truth): ii, LB(Label): i. (b) GT: x, LB: ii. (c), (d) are noisy pictures.

2 Competition description

The Data-Centric AI competition setting aims to improve dataset given a fixed model architecture, which is the opposite of other competitions that aims to find a suitable model (algorithm) for a given dataset. The fixed model is a modified ResNet-50 (He et al. [2016]) whose input is of size 32x32. The dataset has train-validation splits, which totals to roughly 3,000 hand-written Roman numerals. The train split is unbalanced, and some data are perturbed by noise and wrong labels as in Figure 1. Another split called "label book" consists of 52 samples that are representative of each class and can be used in place of a small test set. The model weights are selected with the best validation accuracy in 100 epochs. At the time of the competition, the actual test dataset was not made public. In this setting, we have to somehow construct no more than 10,000 train-validation samples from the original dataset. The goal of the competition is to improve test accuracy with the new dataset under fixed model constraints.[1] In other words, we need to construct a new dataset that can help the model generalize better.

3 Methodology

Although we focus on the competition, we concentrated on domain-agnostic techniques and relied only on the given data (i.e. we do not consider generative or collection technique) to be applied to more general tasks.

First, we considered given dataset as training data and label book as "clean" validation data. After examining the data, we identified the following challenges: (1) many training data points were perturbed by noise in input and label (2) scarcity of training data (3) imbalance of training data (4) scarcity of "clean" validation data, i.e. label book. We organized components to solve these problems.

We define a training data point $z_i^t := (x_i^t, y_i^t) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the label space. Denote the training set as $\mathcal{D}^t := \{z_i^t\}_{i \in [|\mathcal{D}^t|]}$ and validation set with data point z_j^v as $\mathcal{D}^v := \{z_j^v\}_{j \in [|\mathcal{D}^v|]}$. We denote per-sample loss given model parameter θ as $\ell(z; \theta)$. The details of our method are as follows.

3.1 Overall pipeline

The overall pipeline to solve the previous problems is as follows. First, we computed data value or influence(Section 3.2) for deleting negatively affecting data on validation loss. Then, we applied data augmentation(Section 3.4). However, this augmented dataset still had mislabeled and noisy data because the initial cleansing did not fully remove the undesirable points. Hence, we trained the model using contrastive learning and conduct secondary cleansing(Section 3.3) from there. For this part, we set the image size to 64x64 in order not to lose too much information on the image.

In addition, we concentrate on some edge cases and apply more data augmentation for a more compact distribution of data representation. Finally, we apply cleansing again. We alternated cleansing and augmentation of this pipeline for a number of times to obtain a full training data. Our final pipeline is in Figure 2. Moreover, we used the given model architecture from the competition to provide the same inductive bias on the dataset.

3.2 Data cleansing via data valuation

We began by addressing training data points that were negatively impacting the model generalization performance. Rather than manually cleaning the data, our methodology focused on utilizing influence

[1] More details in https://worksheets.codalab.org/worksheets/0x7a8721f11e61436e93ac8f76da83f0e6

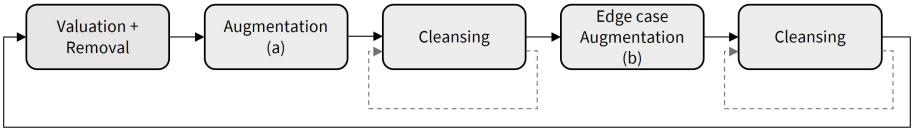


Figure 2: Our overall pipeline for get new dataset. First, we remove negative influence data after which we apply Faster AutoAugment as in (a). Next, we conduct contrastive learning for cleansing. The dashed line means several iterations. (b) is concentrated augmentation for edge case. Finally, do one more cleansing operation. We repeat this process for get full training dataset.

function (IF) (Koh and Liang [2017]). The idea behind IF is to "upweight" a training point to measure its influence at inference. We define an upweighted loss on the i-th training point and optimized model parameter θ_i^* as follows:

$$\mathcal{L}_t^{up}(z_i^t;\theta) = \frac{1}{|\mathcal{D}^t|}\sum_{k=1}^{|\mathcal{D}^t|}\ell(z_k^t;\theta) + \varepsilon_i\ell(z_i^t;\theta), \qquad \theta_i^* := \arg\min_\theta \mathcal{L}_t^{up}(z_i^t;\theta) \tag{1}$$

where ε_i is small weight on i-th train data point. Based on Equation 1, we compute the training data influence on the validation loss by IF:

$$\text{IF}(z_i^t, z_j^v) = -\frac{1}{|\mathcal{D}^t|}\frac{d\ell(z_j^v;\theta_i^*)}{d\varepsilon_i}\Bigg|_{\varepsilon_i=0}, \tag{2}$$

which means the change in the validation loss of the j-th validation data point by upweighting the i-th training point. In order to circumvent the time complexity of computing the inverse Hessian, we employed HyDRA(Chen et al. [2021]) to approximate the influence via hyper-gradient. Since the validation dataset (i.e. label book) was small, we filtered out data points in a conservative manner, excluding only those with large negative influences.

For more details, we calculate the influence of each training data point for each validation data point via Equation 2. Next, we calculate the minimum (min) and standard deviation (std) of all training data influence for each validation point. And remove them if influence less than min + std. We found that the model can be overfitted to the label book if we set the influence criterion too large.

3.3 Data cleansing via contrastive learning

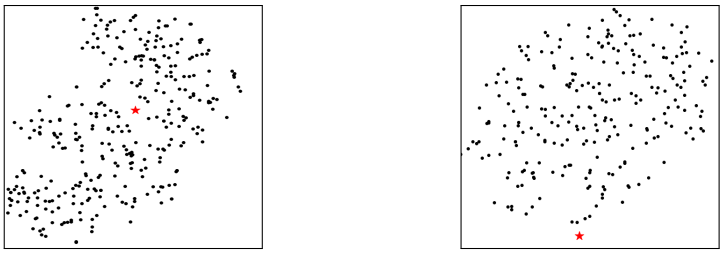
As a second step, for deleting or relabeling obviously perturbed data in pixels and label, we employed supervised contrastive learning with Siamese network[2] trained on the dataset obtained so far. This was trained to distinguish between data of the same class and data of another class. In order to make paired examples for contrastive learning, we applied shear, inversion, shift, rotation, zoom and Gaussian noise augmentation.

Using the learned representation, we could visualize the dataset by projecting their feature in 2-D space via t-SNE (Van der Maaten and Hinton [2008]), and identified the data points that were obviously mislabeled or noise. Thus, using the k-nearest neighbor distances and labels, we fixed the label if it was obviously a labeling error (all neighbors having the same but a different label as in Figure 3.(a)), or dropped the data point if it didn't obviously belong to a cluster (the closest neighbor being far from the data point or it is around a different class cluster as in Figure 3.(b)).

3.4 Data augmentation

We employed a augmentation technique, Faster AutoAugment (FAA, Hataya et al. [2020]) to address training data scarcity. Since train data is not enough, we thought that the classifier would not learn properly because the middle of the data distribution may not be filled. From this point of view, we decided to use FAA which fills in the missing data points. In order to make the augmentation fit the gray scale image setting of the competition, only shear, inversion, translation, rotation, and Gaussian noise were used for the policy search space. In this process, we balanced the number of samples in each class to solve the imbalance.

[2]We modified https://keras.io/examples/vision/siamese_contrastive/



(a) (b)

Figure 3: Example of visualization with learned representation. Each red star is a representative sample of a different class from the cluster. (a) shows a mislabelled point that is placed within the cluster where all its neighbors are of different labels. On the other hand, (b) shows a noisy data that lying at the boundary of the different class cluster.

In addition, we conduct further edge case augmentations. For this, we retrain the model with augmented and cleaned dataset. After that, we project the learned penultimate features onto 2D space with t-SNE from which we identified data points that obviously belong to a cluster but far from the closest same-class neighbor as edge-case data point. On such data points, we applied further data augmentation using inversion, shift, zoom and rotation.

4 Results

Method	Accuracy
Full Pipeline (ours)	**0.84711**
HyDRA, FAA, Inversion	0.82603
HyDRA, FAA	0.80950
HyDRA, Random Augmentation	0.75165
HyDRA	0.67562
Baseline	0.64463

Table 1: Score for our pipeline component transformations

Table 1 presents performance for our pipeline component transformations. We conducted the competition with subtle changes based on the our pipeline and report the accuracy on the leaderboard[3]. Each of them is a combination of our components. Cleansing was included in all combinations except for the Baseline (no modification of the data). Random Augmentation means randomly augmented instead of FAA, and inversion means pixel inversion augmentation to stabilize the representation instead of augmenting edge cases.

As a result, we were able to reconstruct data that had a good impact on performance with small and noisy dataset. At the time of submission, because the actual test set was not disclosed and limit of submission in the competition, the performance of each component was not systematically conducted and not optimized. Hence, there is a room for performance improvement. Finally, we got 84.711% accuracy, taking the 6th place(except duplicated ranks, about 1.1% difference from 1st place). Our approach received the Honorable Mention in the Innovative category by our pipeline.

5 Conclusion

Even though our approaches were not fully developed and were mostly manual at the time of submission, we confirmed the feasibility of such pipeline with the Data-Centric AI competition. For future work, we are going to improve this method with a more algorithmic approach for real-world applications on a more optimized automated pipeline toward data-centric AI.

[3]Leaderboard can be found at `https://https-deeplearning-ai.github.io/data-centric-comp/`

Acknowledgments

This work was supported by the National Research Foundation (NRF) of Korea (NRF-2019R1A2C1087634), and the Ministry of Science and Information communication Technology (MSIT) of Korea (IITP No. 2020-0-00940, IITP No. 2017-0-01779 XAI and IITP No. 2019-0-00075, Artificial Intelligence Graduate School Program(KAIST))

References

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

Yuanyuan Chen, Boyang Li, Han Yu, Pengcheng Wu, and Chunyan Miao. Hydra: Hypergradient data relevance analysis for interpreting deep neural networks. *arXiv preprint arXiv:2102.02515*, 2021.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *European Conference on Computer Vision*, pages 1–16. Springer, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pages 3763–3772. PMLR, 2019.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32:6665–6675, 2019.

Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*, pages 1863–1873, 2020.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 501–509, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pages 10842–10851. PMLR, 2020.