RESEARCH ARTICLE

Mitigating carbon footprint for knowledge distillation based deep learning model compression

Kazi Rafat[1], Sadia Islam[1], Abdullah Al Mahfug[1], Md. Ismail Hossain[1], Fuad Rahman[2], Sifat Momen[1], Shafin Rahman[1]*, Nabeel Mohammed[1]

1 Apurba NSU R&D Lab, Department of Electrical and Computer Engineering North South University, Dhaka, Bangladesh, **2** Apurba Technologies, Dhaka, Bangladesh

* shafin.rahman@northsouth.edu

Citation: Rafat K, Islam S, Mahfug AA, Hossain M. I, Rahman F, Momen S, et al. (2023) Mitigating carbon footprint for knowledge distillation based deep learning model compression. PLoS ONE 18(5): e0285668. https://doi.org/10.1371/journal.pone.0285668

Editor: Süleyman Eken, Kocaeli Üniversitesi: Kocaeli Universitesi, TURKEY

Received: January 17, 2023

Accepted: April 27, 2023

Published: May 15, 2023

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pone.0285668

Data Availability Statement: All data and computational codes are released in the public GitHub repository (https://github.com/King-Rafat/STKD_CFMitigation).

Abstract

Deep learning techniques have recently demonstrated remarkable success in numerous domains. Typically, the success of these deep learning models is measured in terms of performance metrics such as accuracy and mean average precision (mAP). Generally, a model's high performance is highly valued, but it frequently comes at the expense of substantial energy costs and carbon footprint emissions during the model building step. Massive emission of CO_2 has a deleterious impact on life on earth in general and is a serious ethical concern that is largely ignored in deep learning research. In this article, we mainly focus on environmental costs and the means of mitigating carbon footprints in deep learning models, with a particular focus on models created using knowledge distillation (KD). Deep learning models typically contain a large number of parameters, resulting in a 'heavy' model. A heavy model scores high on performance metrics but is incompatible with mobile and edge computing devices. Model compression techniques such as knowledge distillation enable the creation of lightweight, deployable models for these low-resource devices. KD generates lighter models and typically performs with slightly less accuracy than the heavier teacher model (model accuracy by the teacher model on CIFAR 10, CIFAR 100, and TinyImageNet is 95.04%, 76.03%, and 63.39%; model accuracy by KD is 91.78%, 69.7%, and 60.49%). Although the distillation process makes models deployable on low-resource devices, they were found to consume an exorbitant amount of energy and have a substantial carbon footprint (15.8, 17.9, and 13.5 times more carbon compared to the corresponding teacher model). The enormous environmental cost is primarily attributable to the tuning of the hyperparameter, Temperature (τ). In this article, we propose measuring the environmental costs of deep learning work (in terms of GFLOPS in millions, energy consumption in kWh, and CO_2 equivalent in grams). In order to create lightweight models with low environmental costs, we propose a straightforward yet effective method for selecting a hyperparameter (τ) using a stochastic approach for each training batch fed into the models. We applied knowledge distillation (including its data-free variant) to problems involving image classification and object detection. To evaluate the robustness of our method, we ran experiments on various datasets (CIFAR 10, CIFAR 100, Tiny ImageNet, and PASCAL VOC) and models (ResNet18, MobileNetV2, Wrn-40-2). Our novel approach reduces the environmental costs

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

by a large margin by eliminating the requirement of expensive hyperparameter tuning without sacrificing performance. Empirical results on the CIFAR 10 dataset show that the stochastic technique achieves an accuracy of 91.67%, whereas tuning achieves an accuracy of 91.78%—however, the stochastic approach reduces the energy consumption and CO_2 equivalent each by a factor of 19. Similar results have been obtained with CIFAR 100 and TinyImageNet dataset. This pattern is also observed in object detection classification on the PASCAL VOC dataset, where the tuning technique performs similarly to the stochastic technique, with a difference of 0.03% mAP favoring the stochastic technique while reducing the energy consumptions and CO_2 emission each by a factor of 18.5.

Introduction

Deep learning methods are making significant advancements in solving problems deemed infeasible after years of tremendous effort by the artificial intelligence (AI) community [1]. The success obtained by the deep learning techniques has come in different substantial areas of AI such as object detection [2–5], image classification [6–8], natural language processing (NLP) [9, 10], medical imaging and processing [11–13] etc. Typically, models are fine-tuned to increase performance by a small margin to achieve state-of-the-art results. However, the efficacy of deep learning models endure substantial energy costs and produces a considerable carbon footprint that can contribute to climate change that is harmful to life on this earth. We know that Carbon dioxide (CO_2), a greenhouse gas, is a major contributor to global warming [14] that has a direct detrimental effect on agriculture [15], forestry [16, 17], species loss [18], sea-level rise [19], electricity demand [20], human amenity, human morbidity, natural disasters [21], construction, water supply, urban infrastructure, infectious diseases [22] and so on. Nevertheless, according to Strubell et al. [23], a basic NLP pipeline with tuning produces over twice the amount of CO_2 produced by the average American in one year. Being in oblivion, deep learning researchers, focus more on creating better models and prioritize model performance over the society and the environment. In this paper, we investigate the carbon footprint associated with a deep learning strategy called knowledge distillation and suggest a plausible solution to mitigate this problem.

Knowledge distillation is a well-known tool for model compression. Deep learning models are generally large architectures containing millions of parameters (weights). Model compression aims to design a relatively lighter version of the original large version without compromising the overall performance [24]. From the literature, larger and lighter models are called the teacher and student models, respectively [25]. Following the knowledge distillation process, a teacher model shares its knowledge with student models so that the students not only learn from the dataset but also directly from the teacher. In many real-life applications, especially in mobile or edge devices with computational and hardware constraints, deploying the lighter student model (instead of the larger teacher model) can be a pragmatic solution. At present, Hinton's knowledge distillation [26] can be considered the most widely used model compression technique due to its efficacy and simplicity. The distillation process softens output logits of the teacher model to transfer information from the teacher to the student model. The student model uses the logit distribution of the teacher model across all training instances as an altered training set. Soft outputs of the teacher model act as the ground truth (supervision) while learning the student model. This auxiliary information provided by the teacher (contained in the soft outputs) is known as dark knowledge [26, 27]. Dark knowledge also contains

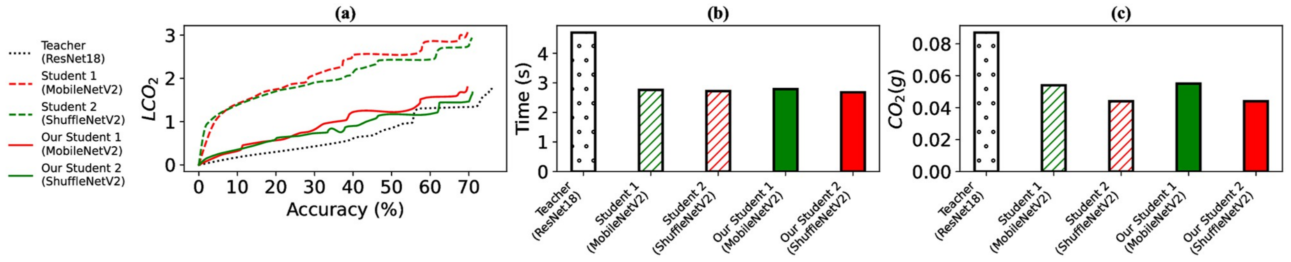


Fig 1. Illustration of carbon footprints used by different deep models while **(a)** training on CIFAR 100 (in log scale) and **(b-c)** inferring on evaluation set. ResNet18 is a deeper model with 11.2M parameters, resulting in higher inference time (4.7 sec.) and CO_2 emission (0.087 g). To minimize this, using ResNet18 as a teacher, we train two student models, MobileNetV2 (student 1) and ShuffleNetV2 (student 2), following the traditional KD process. This training costs significant carbon footprints (red and green dashed curves in **(a)**) with an accuracy increment from learning the teacher model (black dotted curve in **(a)**). However, as expected, both students consume less time and CO_2 during inference (red and green shaded bars in **(b)** and **(c)**). We aim to reduce the training cost and CO_2 production of the KD process while using the same students (red and green solid curves in **(a)**) and maintain similar accuracy and inference costs (solid red and green bars in **(b)** and **(c)**) in comparison with the costly KD training.

https://doi.org/10.1371/journal.pone.0285668.g001

helpful information of class similarity in a dataset that is helpful for the model to learn [26]. For proper knowledge transfer from the teacher to the student, the softening process requires tuning a hyperparameter named Temperature, τ. Hinton et al. [26] used a τ value between 2 and 20 inclusively. However, τ can still be increased to test with softer outputs. An improper value of τ produces inefficient models with a lackluster performance. Therefore, to attain a suitable student model, knowledge distillation frameworks must iterate several times with different values of the hyperparameter τ. Preparations and tuning around such frameworks raise substantial financial and environmental costs by engaging longer training time and higher carbon footprint. In this paper, we estimate the overhead of existing methods and propose a mitigation process by using a stochastic process related to a hyperparameter instead of a grid search [28] (see Fig 1).

To tackle these problems, this paper proposes a stochastic technique of assigning the hyperparameter τ. This process of assigning a stochastic temperature facilitates training while drastically reducing costs and preserving the performance of the assembled student model. This system of training a model compression system can achieve comparable performance to a fine-tuned system while reducing the time, power costs, and carbon footprint. In summary, our contributions are as follows:

- We investigate the environmental costs (carbon footprints) for deep learning model compression using knowledge distillation. It establishes that the cost is not insignificant, and researchers should be aware of this matter considering the ongoing climate change movement.
- We propose a stochastic approach to mitigate the carbon footprint issues related to the knowledge distillation process. It helps to produce satisfactory performance (compressed model without compromising accuracy), reducing energy consumption. One can easily integrate the method with any knowledge distillation pipeline without high overhead costs.
- We perform extensive experiments on different student-teacher model combinations based on ResNet18 [29], MobileNetV2 [30], and ShuffleNetV2 [31] architectures. We report results on both object recognition and detection problems using CIFAR10 [7], CIFAR 100 [7], Tiny Imagenet [32] and PASCAL VOC [33, 34] datasets. For all experiments, we compare carbon footprints (GFLOPs count in millions, energy consumption in kWh, and CO_2 equivalent in grams) between existing and our proposed methods.

Related works

Energy cost for deep learning

Recent technological advancements in hardware, model development, and training frameworks have stimulated the rise of highly accurate deep learning models [9, 35–37]. Such models, however, are expensive in terms of computational resources and energy consumption, resulting in a massive increase in their carbon footprint. Recent studies have investigated energy usage and the environmental impact of deep learning models to raise awareness regarding this situation. Strubell et al. [23] deduced that training large NLP frameworks trends towards a hefty carbon footprint that is detrimental to the environment. Another study [38] discovered that hyper-parameter tuning of large frameworks has a significant carbon footprint, which has a negative impact on climate change. However, such factors are rarely considered because most papers published in major venues aim for accuracy rather than computational efficiency [39]. Parcollet and Ravanelli [40] estimated a huge carbon footprint related to training speech recognizers. Tamburrini [41] discussed the ethical implications for AI researchers, manufacturers, and industries. Given the significance of this problem, very few studies have been aimed at mitigating this issue. Schwartz et al. [39] identified the primary reasons contributing to a high carbon footprint and also proposed the efficiency metrics in deep learning research for proper energy utilization. Anthony et al. [42] also tackled this problem and developed a tool to track and estimate the carbon footprint produced by deep learning frameworks. This study aims to assess student-teacher frameworks' carbon footprint for model compression.

Knowledge distillation methods for model compression

Buciluǎ et al. [43] first introduced the term 'model compression' to compress large complex ensembles into smaller and faster models without significant performance loss. Knowledge distillation (KD) is a popular strategy to address the model compression problem. KD trains a smaller network with the help of larger models and ground labels. Lei and Caruana [44] demonstrated that shallow feed-forward nets could learn the complex functions previously learned by deep nets and achieve performances previously only achievable with deep models. Later, Hinton et al. proposed a KD-based model compression method [26] that distills a teacher's knowledge using a special loss by softening the outputs of the teacher to a student model, which is widely used for compression tasks. Further improvements in this line of investigation followed Hinton's distillation process. Liu et al. [45] improved novelty in Hinton's distillation approach by distilling pixel-wise and structure distillation schemes. Park et al. [46] have resorted to a structured comparison of the teacher and student to improve Hinton's distillation. In this paper, we investigate knowledge distillation in object classification and detection in the context of model compression.

Applications of KD

Recently, knowledge distillation has been successfully applied to solve complex machine-learning problems, mainly image classification. Li and Hoem [47] used knowledge distillation to make a unified vision system that utilizes the knowledge in models to add new capabilities to the system without requiring training data. Liu et al. [48] focused knowledge distillation on Multi-Label Classification, which focuses on creating a Weakly Supervised Model (WSM) to distill its knowledge for better classification. Wang and colleagues [49] put to use knowledge distillation to explain and interpret image classifiers by developing smaller image classifiers that are easily explainable. Peng et al. [50] utilizes correlation between images to distill knowledge for improvement in image classification. Knowledge distillation and model compression are currently being used for object detection, as models tend to be large and require a costly runtime. Chen et al. [51] employed knowledge distillation to make efficient object detection models with a similar performance by innovating model compression to suit object detection. Li and colleagues [52] distilled models for improving a photon-limited object detection framework. Guo et al. [53] showcases a method of online knowledge distillation where all DNNs are considered as students for a more effective distillation for object detectors. Dai and his [54] team used an alternative form of knowledge distillation for object detection that uses three knowledge distillations for proper knowledge transfer. The routine unavailability of training data due to privacy issues has resulted in a recent increase in the popularity of data-free knowledge distillation, where models are trained through distillation without the training data. Chawla et al. [55] utilizes knowledge distillation for data-free object detection such that the generated data enables better distillation of knowledge. Zhang and colleagues [56] developed a generator network for producing artificial training data for image super-resolution using progressive knowledge distillation. In their work, Yin and team [57] discussed their data production technique, Deep Inversion, which produces training data from a trained network by inverting the model without prior knowledge of the data. Choi et al. [58] utilizes meta-data from the teacher network to generate artificial data using a generator that was not trained on training data for knowledge transfer to a smaller network. Note that knowledge distillation is also readily used in n-shot learning techniques. Nayak et al. [59] proposed techniques that allow zero-shot knowledge distillation without data samples or meta-data. Michaeli and Storkey [60] also proposed a novel method to transfer knowledge to students using a generative adversarial network for a zero-shot knowledge transfer. In this paper, we propose a stochastic approach of assigning temperature to knowledge distillation frameworks to eliminate the hefty requirement of the tuning hyper-parameters temperature.

Methodology

Recently, various model compression techniques have been proposed for deep learning, primarily utilizing information from the teachers to train a smaller student better (knowledge distillation) [26, 43, 44], modifying networks to reduce the model size (pruning) [61–63] or trading precision for a smaller size (quantization) [64, 65]. In this section, we define the problem, the drawbacks of the current solution, a proposal for improvements, and relate carbon footprint issues with AI ethics.

Problem formulation

Suppose, $\mathcal{F}_m(\boldsymbol{X}; \boldsymbol{W_m})$ is a deep neural network model where $\boldsymbol{X}$ and $\boldsymbol{W_m}$ represent the input image and model parameters/weights, respectfully. Let us consider, number of parameters, $|\boldsymbol{W_m}|$ is relatively high, meaning a deeper model which obviously performs well in a recognition or detection tasks. However, computation cost during inference is high because of its larger size. To reduce the computational cost, our aim is to design a compressed model, $\mathcal{F}_s(\boldsymbol{X}; \boldsymbol{W_s})$ where, number of parameters $|\boldsymbol{W_s}|$ is significantly smaller than $|\boldsymbol{W_m}|$, i.e., $|\boldsymbol{W_s}| << |\boldsymbol{W_m}|$ but performance (accuracy or mAP) of both $\mathcal{F}_m(:; \boldsymbol{W_m})$ and $\mathcal{F}_s(:; \boldsymbol{W_s})$ models are similar. The architecture of $\mathcal{F}_s(:; \boldsymbol{W_s})$ and $\mathcal{F}_m(:; \boldsymbol{W_m})$ can be same or different. To train $\mathcal{F}_s(:; \boldsymbol{W_s})$, we can use logits of the model $\mathcal{F}_m(:; \boldsymbol{W_m})$ for knowledge transfer. Therefore, in this setup, $\mathcal{F}_m(:; \boldsymbol{W_m})$ and $\mathcal{F}_s(:; \boldsymbol{W_s})$ are called teacher and student models, respectively.

Solution using Knowledge distillation (KD) [26]

KD is a well-known tool to compress a possibly larger (teacher) model, $\mathcal{F}_m(\boldsymbol{X}; \boldsymbol{W_m})$ to a smaller (student) model $\mathcal{F}_s(\boldsymbol{X}; \boldsymbol{W}_s)$. Here, the input space X belongs to label space $\boldsymbol{y} \in \mathbb{R}^n$, where n represents input the number of classes available in the dataset. We assume a supervised classification setting where model outputs are $\boldsymbol{a_m} = \mathcal{F}_m(\boldsymbol{X}; \boldsymbol{W_m}), \boldsymbol{a}_s = \mathcal{F}_s(\boldsymbol{X}; \boldsymbol{W}_s); \ \{\boldsymbol{a}_m, \boldsymbol{a}_s \in \mathbb{R}^n\}$ before activation such that $\boldsymbol{a_m}$ and $\boldsymbol{a_s}$ represent the un-activated output of teacher and student models, respectively. The student model outputs are activated using the softmax function [66] to produce a hard embedding $\boldsymbol{h_s}$ such that $\boldsymbol{h_s}(i) >> \boldsymbol{h_s}(j)$, $\{1 <= i, j <= n, i \neq j\}$ where the logits are probability distributions $\sum_{i=1}^{n} \boldsymbol{h}_s(i) = 1, \boldsymbol{h}_s(i) \in \mathbb{R}^+$. However, transfer of knowledge is done through proper softening of the output of the teacher $\boldsymbol{a_m}$ and student model $\boldsymbol{a_s}$, which is done by a modified softmax activation containing a hyper-parameter τ:
$\boldsymbol{o}_z = \sigma(\boldsymbol{a}_z) = \frac{e^{\boldsymbol{a}_z(i)/\tau}}{\sum_{j=1}^{n} e^{\boldsymbol{a}_z(i)/\tau}}$ where, $\ \boldsymbol{a}_z(i), \boldsymbol{a}_z(j) \in \boldsymbol{a}_z \subset \mathbb{R}, \ |\boldsymbol{a}_z| = n, \ 1 <= i, j <= n$. Here the variable τ is a hyper-parameter crucial for softening of logits $\boldsymbol{o_s}$, which theoretically ranges from $1 < \tau < \infty$. The activation provides softened outputs (probability distribution) for $\boldsymbol{o_s}$ (student) and $\boldsymbol{o_m}$ (teacher) using $z = s$ or $z = m$. Here, $\boldsymbol{o}_s = [\boldsymbol{o}_s(1), \boldsymbol{o}_s(2), \boldsymbol{o}_s(3), \ldots, \boldsymbol{o}_s(n)]$, where, $\sum_{i=1}^{n} \boldsymbol{o}_s(i) = 1, \ \boldsymbol{o_{sn}} \in \mathbb{R}^+$ and $\boldsymbol{o_m} = [\boldsymbol{o_m}(1), \boldsymbol{o_m}(2), \boldsymbol{o_m}(3), \ldots, \boldsymbol{o_m}(n)]$, where, $\sum_{i=1}^{n} \boldsymbol{o_m}(i) = 1, \ \boldsymbol{o_m}(n) \in \mathbb{R}^+$.

The acquisition of soft labels helps utilize the dark knowledge stored in the teacher model for proper knowledge transfer to the student model. The outputs $\boldsymbol{h_s}$, $\boldsymbol{o_s}$, $\boldsymbol{o_m}$ and label $\boldsymbol{y}$ are used to provide gradient information for the student to learn better using the distillation loss which is back-propagated to update model parameters. The distillation loss ($\mathcal{L}_{DL}$) consists of the weighted union of the softmax Cross Entropy ($\mathcal{L}_{CE}$) [67] and the Kullback-Liebler Divergence ($\mathcal{L}_{KLD}$) [68] loss with an added weight.

$$\mathcal{L}_{CE}(\boldsymbol{y}, \boldsymbol{h}_s) = -\sum_{c=1}^{n} \boldsymbol{y}(c) \log \boldsymbol{h}_s(c), \qquad \mathcal{L}_{KLD}(\boldsymbol{o}_s, \boldsymbol{o_m}) = \tau^2 \sum_{c=1}^{n} \boldsymbol{o_m}(c) \log \frac{\boldsymbol{o}_s(c)}{\boldsymbol{o_m}(c)} \tag{1}$$

Knowledge distillation varies from a traditional classification because of $\mathcal{L}_{KLD}$, which facilitates the student to learn. The hyper-parameter τ usually gets a value within the range $\tau = 2$ to 20. τ contributes to softmax distributions with different levels of softness. Proper softening of the logits enables better distribution of information to the student, S. As a result, the hyperparameter temperature is essential in knowledge distillation for proper knowledge transfer. For instance, a larger model might require more softening, i.e., a higher value of τ is expected. Additionally, knowledge distillation requires another hyper-parameter α which is used to assign weights to the Cross-Entropy and Kullback-Liebler loss functions for proper parameter tweaking by gradient descent. Fine-tuning is required in some cases for perfect model distillation. Finally, the distillation loss is formulated as follows:

$$\mathcal{L} = \underbrace{\alpha \ \mathcal{L}_{KLD}(\boldsymbol{o}_s, \boldsymbol{o}_t)}_{\tau=2 \ \text{to} \ \infty} + \underbrace{(1-\alpha) \ \mathcal{L}_{CE}(\boldsymbol{y}, \boldsymbol{h}_s)}_{\tau=1} \tag{2}$$

The process of knowledge distillation is briefly shown schematically in Fig 2.

Issues with knowledge distillation

The Kullback-Liebler loss provides additional gradient information to the distillation loss that transfers a teacher's knowledge to a student. Properly softening the hyper-parameter Temperature, $\boldsymbol{\tau}$, is required to produce the softening effect that introduces dark knowledge for the

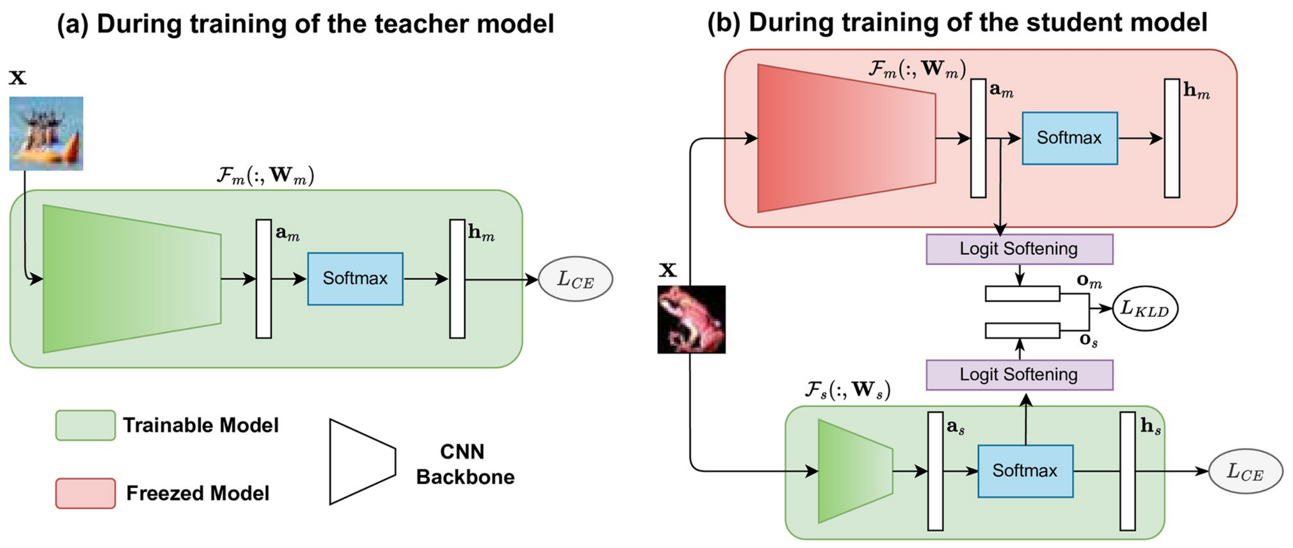


Fig 2. Block diagrams of KD architectures while training teacher, $\mathcal{F}_m(:, W_m)$ and student, $\mathcal{F}_s(:, W_s)$ models. (a) Given input X, trainable $\mathcal{F}_m(:, W_m)$ (indicated as green) learns to produce logits h_m after a softmax activation. Cross-Entropy loss ($\mathcal{L}_{CE}$) is used to train the teacher model. **(b)** Trainable student model, $\mathcal{F}_s(:, W_s)$ (indicated as green) leanrs from a frozen (indicated as red) teacher, $\mathcal{F}_m(:, W_m)$. The teacher and student produce unactivated logits a_m and a_s, respectively. a_s are activated using a softmax containing the hyperparameter τ producing soft logits o_m. Similarly, o_s is produced for the teacher. The soft logits are used to calculate KLD loss ($\mathcal{L}_{KLD}$). The hard logits from the student h_s are further used to calculate $\mathcal{L}_{CE}$. $\mathcal{L}_{KLD}$ acts as additional supervision for the student to learn better.

https://doi.org/10.1371/journal.pone.0285668.g002

student to consume. An unsuitable value of τ might harm students' training. In contrast, Fig 3(a) shows that adjusting/tuning a temperature for knowledge distillation may lead to performance enhancement. Therefore, hyper-parameter τ requires fine-tuning through grid search [69]. Fine-tuning such a hyper-parameter requires additional time and bears huge computation and energy costs. To facilitate this tuning, KD training may produce a huge carbon footprint, especially for a vast dataset. In this way, model tuning contributes to climate change and becomes a concern for ethical AI.

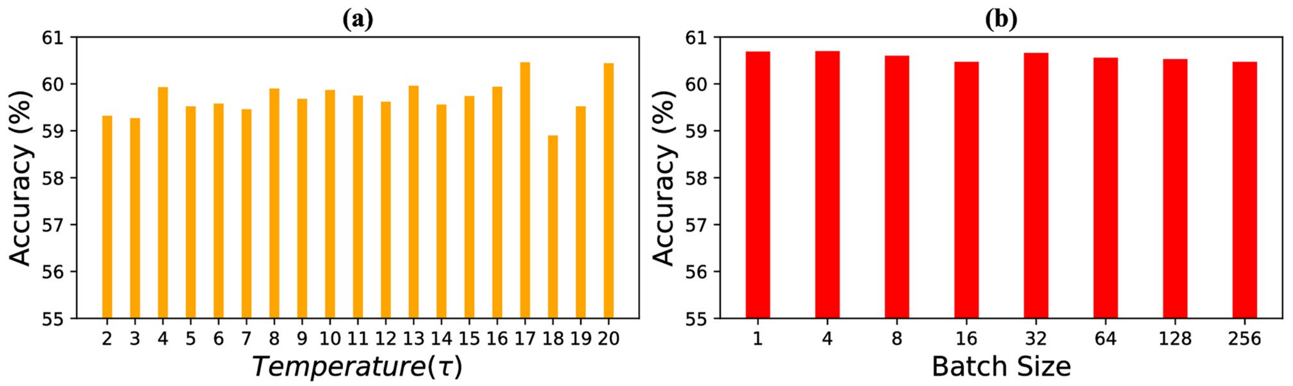


Fig 3. Impact of using different τ for ResNet18 (teacher) and MobileNetV2 (student) models using **(a)** Tiny ImageNet datasets. We notice a significant performance variance across different τ values. **(b)** The impact of using different batch sizes for our proposed stochastic solution. Similar performance across different batch sizes shows that our proposal does not depend on training batch sizes.

https://doi.org/10.1371/journal.pone.0285668.g003

Carbon footprint for knowledge distillation

In this subsection, we discuss the carbon footprint estimation process used in this paper and analysis the energy consumption of the traditional KD process.

Measurement criteria. We use three criteria [42] to measure the carbon footprints of the KD process: FLOPs count in millions, energy consumption in kWh, and CO_2 equivalent in grams. Now, we briefly summarize those criteria below.

FLOPs count. FLOPs (Floating-point Operations) estimate the number of additions and multiplications a model performs for a singular instance that the model receives. FLOPs count for a model is a vital efficiency measure since it showcases the number of calculations needed for a prediction. It also determines the sophistication of the model and the time it consumes. Large models tend to make more calculations, i.e., higher FLOPs count.

Energy. Energy consumption directly correlates an experiment with the amount of carbon footprint it produces. The power collected in the study is the total power of the computer components, mainly CPU, GPU, and DRAM. The total power consists of the overall power used by the computer components and not the power consumed by a deep learning experiment. As a result, energy/power consumption may vary for a similar experiment. Accurately attaining dynamic power or power for particular work is challenging and, at times, infeasible. Due to limited components tracked for power, the consumed is multiplied by the Power Usage Effectiveness (PUE). The PUE is the ratio of the total power used by the data center to the energy used to make a computation. We use PUE = 1.67, the global average for the year 2019 [70], because of the infeasibility of calculating the PUE in the locality where experiments are conducted.

First, the instantaneous power, p_i, or the power consumed per second, is measured in kWh for a duration of one training epoch. We denote t_e as total number of epochs. Using the instantaneous power, the average power p_{avg}, can be calculated as follows:

$$p_{avg}(kWh) = \frac{1}{t_e}\sum_{i=1}^{t_e} p_i \tag{3}$$

Next, the total power consumed by the computer components for one singular epoch e on device d is summed over the total number of training epochs E. The consumption for all the computational devices D, using the average power p_{avg}, requiring t_e time for one training epoch, and multiplied with the PUE shown in Eq. below:

$$p_t(kWh) = PUE \times \sum_{e\in E}\sum_{d\in D} p_{avg} \times t_e \tag{4}$$

CO_2 equivalent. To estimate the carbon emission, we calculate CO_2 equivalent (c_t) by the product of the carbon intensity c_i of the area where the experiment is being carried out, and the energy consumed p_t, of that experiment.

$$c_t(g) = p_t \times c_i \tag{5}$$

Energy estimation. In Table 1, we report the carbon footprints of the KD process. Large learning models (like the teacher) have a better learning capacity (Acc. 95%) due to many trainable parameters. However, such models usually have a large model size and take longer time for prediction, producing a vast carbon footprint. In contrast, lightweight models (like the student) are relatively small in architecture and faster during inference. However, such models lack performance (Acc. 90.6%) compared to large models. Model compression techniques such as KD are used to train lightweight models to mimic large models' performance.

Table 1. Carbon footprints of Teacher (ResNet18) and Student (MobileNetV2) model before KD (top two rows), traditional KD [26] and `Ours` approach on three different datasets. ↑ (↓) means higher (lower) is better.

	CIFAR 10			CIFAR 10			Tiny ImageNet		
Method	**Accuracy (%) ↑**	**Energy (kWh) ↓**	**CO_2_eq (g) ↓**	**Accuracy (%) ↑**	**Energy (kWh) ↓**	**CO_2_eq (g) ↓**	**Accuracy (%) ↑**	**Energy (kWh) ↓**	**CO_2_eq (g) ↓**
Teacher	**95.04**	0.28	74.25	**76.03**	0.25	65.55	**63.39**	1.43	413.60
Student	90.52	**0.16**	**45.02**	67.80	**0.16**	**44.23**	59.64	**0.68**	**182.53**
KD [26]	**91.78**	3.99	1173.44	**69.70**	3.80	1170.40	60.46	19.09	5595.31
`Ours`	91.67	**0.21**	**61.76**	69.40	**0.20**	**61.60**	**60.53**	**1.02**	**304.88**

https://doi.org/10.1371/journal.pone.0285668.t001

A student model taught using a good teacher model performs surprisingly better when fine-tuned (Acc. 91.8%). However, the KD technique has disadvantages because it consumes about 25 times more energy and produces substantial carbon footprints than training a similar base model (see Table 1). For instance, image classification with KD on a standard Tiny ImageNet dataset using a ResNet18 teacher and a ShuffleNetV2 student produces 5595.31g of CO_2 (see Table 4). It consumes 19.09 kWh of energy for a fully tuned experiment, equivalent to a car traveling 46.5km. Even smaller datasets regularly used as benchmarks, such as CIFAR 10, produce a considerable carbon footprint, about 1170.21g, and consume about 3.99 kWh of power (see Table 4) for a tuned experiment using a prominent teacher, small student combination. Modern usage of knowledge distillation, such as in data-free knowledge distillation, also produces a hefty amount of CO_2 in both data production and model compression. Data-free knowledge distillation using the Contrastive Model Inversion technique produces 1353.7g of CO_2 (see Table 6) on the small and standard CIFAR 10 dataset. It yields a nearly similar amount of CO_2 (1369.46g) (see Table 6) when using Deep Inversion as the data-generating technique. Recently, knowledge distillation is also prevalent in object detection settings. Single-Shot Multibox Detection (SSD) with a VGG16 teacher and MobileNetV2 student consumes 1.69 kWh energy for a perfectly tuned model. It produces 9235.39g of CO_2 (see Table 4), estimated to be equal to a car traveling for 78.66km.

Stochastic solution. As discussed earlier, the KD process is costly because of the hyperparameter τ selection that ranges theoretically $2 <= \tau <= \infty$. We now discuss different strategies to select τ efficiently and propose our recommendation.

Grid vs. random search. Grid search is a hyperparameter tuning process that iterates over every combination of pre-defined hyperparameter values. Its search space is relatively high, which can contribute to carbon footprints significantly. In contrast, random search randomly selects a combination of hyperparameters from a search space. Nonetheless, random search has shown better performances in some cases where increasing the search space finds a better combination when the important hyperparameter is unknown [28]. However, hyperparameter τ controls the level of softening during KD, which is sensitive to performance. As a result, random search is unsuitable in this scenario. τ is usually tuned through costly grid search.

Dynamic temperature, τ. Several works attempt to find dynamic τ for better performance. Jafari et al. [71] use a dynamic temperature, but the training procedure consists of two stages which are computation heavy. It also requires choosing a max temperature at the beginning of the training that can be treated as another hyperparameter. Liu et al. [72] employ dynamic hyperparameters by using another model/network to update parameters according to validation performance which helps to improve accuracy. Li et al. [73] also use a two-layer MLP for the same reason. All these techniques focus on maximizing performance. With the cost of considerable energy for training, they achieve a minuscule rise in performance. We argue that such a process is not green computation friendly.

Our proposal. Deep networks are trained using random batches, which produce hard output vectors ($\boldsymbol{h_m}$ for teacher or $\boldsymbol{h_s}$ for student). Hyperparameter, τ controls the softening of each element inside $\boldsymbol{h_m}$ or $\boldsymbol{h_s}$. Knowledge distillation utilizes dark knowledge for information transfer which is the information in the soft outputs. The logit information is blurred or smoothed out if the temperature is too high. If the temperature is too low, the embedding becomes sparse [74]. As a result, stronger models usually need higher temperatures since they produce sparse logit embeddings (high probability for target class) and vice versa. For better knowledge transfer, this softening should vary from instance to instance. Considering the random instances inside a batch and different batches should produce random levels of embedding sparsity, different τ is expected to match instances' diversity. A constant temperature throughout the training cannot model such randomness during the learning process. Grid/random search considers a single τ for the entire training. Similarly, dynamic τ use the same value from many consecutive epochs. For this reason, they must repeat the train-validation loop employing multiple τ, which increases the training cost. Therefore, we recommend using a random τ for each batch within a range (say [2, 20]). It will also exclude the necessity of tuning τ. The random τ works with random batches since every random batch requires random softening. In Fig 3(b), we show the impact of using different batch sizes considering random τ within [2, 20]. We notice that our stochastic proposal works consistently across batch sizes. It proves the necessity of using random τ during training. Evidently, through empirical evidence, we establish that the stochastic temperature performs similarly or, at times, better in the case of traditional knowledge distillation in comparison to tuning. Along with a similar performance, the stochastic technique is efficient as it produces significantly less carbon footprint and consumes less energy than classical grid-search or random-search (see Table 1).

Considering a sample input with its label, $(\mathbf{X}, y)$ which is an element in dataset D, a student model $\mathcal{F}_s(\boldsymbol{X}, \boldsymbol{W_s})$ produces soft logits $\boldsymbol{o_s}$ from unactivated logits $\boldsymbol{a_s}$. The training objective is to match the soft logits $\boldsymbol{o_s}$ to the soft logits $\boldsymbol{o_m}$, of a large teacher $\mathcal{F}_m(\boldsymbol{X}, \boldsymbol{W_m})$, from the teacher's own unactivated logits $\boldsymbol{a_m}$, softened by a temperature τ, using an altered softmax $\boldsymbol{o_z} = \sigma(\boldsymbol{a_z}) = \frac{e^{\boldsymbol{a_z}(i)/\tau}}{\sum_{j=1}^{n} e^{\boldsymbol{a_z}(j)/\tau}}$ where, $\boldsymbol{a_z}(i), \boldsymbol{a_z}(j) \in \boldsymbol{a_z} \subset \mathbb{R},\ |\boldsymbol{a_z}| = n,\ 1 <= i, j <= n$. Knowledge distillation uses a KL divergence loss as additional information from the teacher given by $\mathcal{L}_{KLD}(\boldsymbol{o_s}, \boldsymbol{o_m}) = \tau^2 \sum_{c=1}^{n} \boldsymbol{o_m}(c) \log \frac{\boldsymbol{o_s}(c)}{\boldsymbol{o_m}(c)}$ (as shown in Eq 1) softened and weighted by τ. Considering the model is learning for ω epochs and with a constant τ as in traditional KD, the teacher output $\boldsymbol{o_m}$, remains the same but the student model output $\boldsymbol{o_s}$ changes to optimize towards $\mathcal{L}_{KLD}(\boldsymbol{o_s}, \boldsymbol{o_m})[\omega_p] < \mathcal{L}_{KLD}(\boldsymbol{o_s}, \boldsymbol{o_m})[\omega_q]$, such that $\omega_p > \omega_q$. In this setting, for the input $\boldsymbol{X}$ the target $\boldsymbol{o_m}$ is constant throughout the training cycle of the student due to a fixed τ. In our proposed stochastic approach, the value of τ is varied stochastically for every batch, thus ensuring that the target $\boldsymbol{o_m}$ for an input sample $\boldsymbol{X}$ remains varies. This variety in $\boldsymbol{o_m}$ may prevent $\mathcal{F}_s(\boldsymbol{X}, \boldsymbol{W_s})$ from learning to optimize from a single $\boldsymbol{o_m}$ and therefore likely providing a regularizing effect. This may eliminate the need to search for a fixed validated τ.

Green computing and ethical AI. Along with its benefits, AI's usage and development pose severe ethical concerns regarding multiple individual/societal and environmental issues. Deep learning often requires the distribution of sophisticated hardware that requires expensive materials to develop that are running out [75]. Moreover, AI and its development require a lot of computation-hungry methods, which consume substantial computational power and lead to the depletion of natural resources and valuable materials. Furthermore, energy consumption is very high and increasing steadily in the case of computer vision and NLP [23, 39]. Additionally, the power consumption for such AI methods adds up to a large global carbon footprint that is highly harmful to the environment and contributes heavily to climate change.

AI development is increasingly unsustainable due to the significant usage of these natural resources. To tackle these problems, developers and researchers should have equitable access to resources, figure out or use techniques concerned with Green AI, and report the cost for training and tuning deep learning models [23]. This paper may assist in developing strategies that deal with making the hyper-parameter tuning in knowledge distillation and adding to Green AI. Here are some recommendations concerning knowledge distillation and green computing: ***(1)*** KD method-related papers should report carbon footprint metrics (FLOPs count, energy consumption, and CO_2 equivalent) regarding model training. ***(2)*** In addition to performance (accuracy/mAP score), the research community should establish some green index to start a healthier competition and promote energy-efficient methods.

Experiment

Setup

Dataset. We investigate our approach based on object recognition and detection tasks. We present a summary of the used dataset in Table 2 and discuss them below.

Image classification setup. The image classification setup consists of a traditional image classification framework. The experiments are carried out on three popular standard classification datasets: CIFAR 10 [7], CIFAR 100 [7] and Tiny Imagenet [32]. The CIFAR 10 and CIFAR 100 datasets contain 50,000 training images and 10,000 validation images, with a 32×32 resolution, over 10 and 100 individual classes respectively. The Tiny Imagenet dataset is made up of 100,000 training images and 10,000 validation images with a resolution of 64×64. This dataset is made up of 200 individual classes. Classes in the datasets are all mutually exclusive.

Object detection setup. For object detection, we follow conventional object detection frameworks that leverage knowledge distillation. We use the combination of the PASCAL VOC 2007 [33] and the PASCAL VOC 2012 [34] dataset for training. The PASCAL VOC 2007 dataset consists of 5011 training/validation images and the PASCAL VOC 2012 contains 11530 training/validation images over the same 20 classes in the PASCAL VOC 2007 dataset. For model evalutation we used the PACAL VOC 2007 test set which consists of 4952 images over 20 different class for evaluating model performance.

Student-teacher models. We used three different learning models, e.g., ResNet18 [29], MobileNetV2 [30] and ShuffleNetV2 [31] for the image classification framework. In Table 3, we include details of those architectures and how we use those in our work. We evaluate these models on different loss functions, datasets, and teacher-student combinations. This includes strong teacher strong student, strong teacher weak student, weak teacher strong student, and weak teacher weak student. In this paper, we follow strong/weak definitions from [76]. Strong models are usually larger than weak models containing many trainable parameters. Because of this, strong models are expected to perform better than weak models. We first train a teacher model, and after that, we train a student model leveraging the output of the teacher model. In a real-life application, a teacher model is supposed to be strong (heavier model) having higher accuracy, and our goal is to train a lighter student model that can perform similarly to the

Table 2. Summary of datasets used in the study.

Dataset	Training Images	Test Images	Classes	Pixel Size	Train Images Per Class	Test images Per Class
CIFAR 10 [7]	50000	10000	10	32x32	5000	1000
CIFAR 100 [7]	50000	10000	100	32x32	500	100
Tiny ImageNet [32]	100000	10000	200	64x64	500	50
PASCAL VOC [33, 34]	16541	4952	20	-	-	-

https://doi.org/10.1371/journal.pone.0285668.t002

Table 3. Information about different architecture models used in this study.

Model	Number of Parameters (M)	Used as Teacher	Used as Student	Used for classification	Used for DFKD	Used for detection
ResNet18	11.2	✓	✓	✓	×	×
ShuffleNetV2	1.5	✓	✓	✓	×	×
MobileNetV2	3.4	×	✓	✓	×	×
Wrn-40-2	2.2	✓	×	×	✓	×
Wrn-16-1	0.2	×	✓	×	✓	×
VGG16 SSD	26.2	✓	×	×	×	✓
MobileNetV2 SSD lite	3.4	×	✓	×	×	✓

https://doi.org/10.1371/journal.pone.0285668.t003

strong teacher model. Therefore, the computational cost/time of inference will reduce in the student model compared to the teacher model.

Data-free knowledge distillation (DFKD). For DFKD, we investigate two prominent DFKD techniques: Contrastive Model Inversion (CMI) [77] and Deep Inversion (DI) [57]. For both CMI and DI, we used Wrn-40-2 [78] as the teacher model to train a Wrn-16-1 [78] student model on synthetic CIFAR 10 and CIFAR 100 data.

Object detection setup. Object detection looks into classifying multiple classes in a particular image and localizing the class area using a bounding box. Recent development in object detection has brought about new state-of-the-art techniques, Single-Shot Multibox Detector (SSD) [3]. As a result, this study looks into using the stochastic technique on SSD using knowledge distillation. For experiments on object detection, we used VGG16 [3, 79] SSD teacher on a MobileNetV2 SSD lite student [3, 30]. The paper looks into the PASCAL VOC 2007 and PASCAL VOC 2012 training subsets as the dataset for training, and we use PASCAL VOC 2007 as the testing set.

Evaluation metric. For classification and detection problems, we evaluate all methods with accuracy and mean average precision (mAP) in percentage, respectively. ***Accuracy***: We employ testing accuracy to gauge the performance of various datasets as an evaluation criterion for classification and data-free knowledge distillation. The percentage (%) of accurately predicted instances in a testing/evaluation set over the total number of instances in the set is used to determine test accuracy. We also report other performance metrics such as Precision, Recall, Kappa, Specificity, Prevalence, and F1 Score to determine the overall performance.

mAP. For detection issues that need classification and localization, mAP is the most well-known evaluation technique. The IoU (Intersection Over Union) metric, which measures the intersection of the labeled localization area and the predicted localization area over the union of the labeled localization area and the expected localization area, is used to assess a model created for a detection problem. We consider the prediction accuracy when the IoU is greater than or equal to a predefined threshold. Then, we can determine the precision (the proportion of positives successfully predicted among all positive predictions) and recall (the ratio of positives correctly predicted among all positives labeled) for every class. This calculation uses the calculated IoU for every instance of that class from the area of the predicted and group truth bounding box. The precision and recall values are then plotted as a function of the IoU threshold. Lower IoU values predict higher positive predictions. However, not all positive predictions are correct. As a result, recall increases, and precision decreases. So the curve gets a downward slope. Integrating the curve returns the area under the precision-recall curve, referred to as AP (Average Precision). The average of the AP across all the classes is called mAP, which is used to evaluate detection models.

Implementation details. For all experiments, we start with a learning rate of 0.1 and use min-batch gradient descent for parameter training. We run knowledge distillation for classification, DFKD, and object detection experiments for 100, 200, and 70 epochs, respectively. The learning rate is divided by 0.2 to decay at the 40^{th}, 60^{th}, and 80^{th} epoch for the classification setup. The learning rate was similarly decayed by 0.2 at 120^{th}, 160^{th}, and 180^{th} epoch for the data-free knowledge distillation setup. We run the models on an Adam optimizer. The data go through some basic augmentation, such as a random horizontal flip, random rotation of 15 degrees, and a random crop of the image size after 4-pixel padding. We implement the models according to their related papers [3, 29–31, 78, 79]. Also, we train teacher models for 100 epochs and then use them during the knowledge distillation process to learn the student model. We train the teacher models using the cross-entropy loss for both the classification, detection, and DKFD frameworks. We used datasets CIFAR 10, CIFAR 100, Tiny ImageNet, and PASCAL VOC to train the teacher models. For training the student model, we use three loss functions to show the extensiveness of the stochastic technique: distillation loss, L2 loss, and DML loss. To train the student models, we use the output of the teacher model on the loss function to teach the student model. For all experiments, we use an α value of 0.95 and a tuned temperature or the stochastic technique when appropriate. We ran experiments on DKFD and object detection knowledge distillation on official implementation [51, 77]. We have used PyTorch (v1.13 and CUDA 11.3) deep learning framework. We have experimented using a workstation with a single NVIDIA RTX 3070 Ti GPU and 32GB of RAM.

Main results

Compared methods. This paper explores three of the most widely used knowledge distillation methods: Hinton's knowledge distillation (HKD) [26], Caruana's model compression (CMC) [44] and the Deep Mutual Learning Loss (DML) [80]. The CMC and DML loss are enhanced with a temperature hyper-parameter such that a tuned temperature produces better performance than the original loss function. The CMC loss calculates the squared difference between the output and the target logits. The CMC loss does not require additional hyperparameters for softening or weight distribution. However, it does work better with a temperature hyperparameter and is prone to outliers. The DML loss calculates the probabilistic differences between the target and output. The DML loss also does not require softening and weight distribution parameters like the CMC loss. However, it does work better with a temperature hyperparameter. Here, we summarise all loss equations together.

- CMC loss: $\mathcal{L}_{CMC}(\boldsymbol{h}_s, \boldsymbol{h}_m) = \frac{1}{N}\sum_{c=1}^{n}(\boldsymbol{h}_s(c) - \boldsymbol{h}_m(c))^2$
- Altered CMC loss: $\mathcal{L}_{CMC}(\boldsymbol{h}_s, \boldsymbol{h}_m) = \frac{1}{N}\sum_{c=1}^{n}(\frac{\boldsymbol{h}_s(c)}{\tau} - \frac{\boldsymbol{h}_m(c)}{\tau})^2$
- DML loss: $\mathcal{L}_{DML}(\boldsymbol{h}_s, \boldsymbol{h}_m) = \sum_{c=1}^{i}\boldsymbol{h}_m(c)\log\frac{\boldsymbol{h}_s(c)}{\boldsymbol{h}_m(c)}$
- Altered DML loss: $\mathcal{L}_{DML}(\boldsymbol{o}_s, \boldsymbol{o}_m) = \sum_{c=1}^{i}\boldsymbol{o}_m(c)\log\frac{\boldsymbol{o}_s(c)}{\boldsymbol{o}_m(c)}$

Classification. Table 4 demonstrates results obtained for image classification standard classification datasets (CIFAR 10, CIFAR 100, and Tiny ImageNet) using a tuned temperature and the stochastic technique. ***(1)*** Performance obtained for the stochastic technique is similar to a tuned temperature for the HKD loss and DML loss. The CMC loss performs better using a tuned temperature than the stochastic temperature. This is because the CMC loss does not use softening to transfer knowledge. Nonetheless, the power consumption and CO_2 production

Table 4. Carbon footprints and performance for different KD approaches. Here, for image recognition, we use ResNet18 (teacher) and MobileNetV2 (student) models. For object detection, we use VGG16 (teacher) and MobileNetV2 (student). We report average results after running the same program five times. ↑ (↓) means higher (lower) is better.

		CIFAR 10				CIFAR 10			
Loss	Method	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓
HKD	[26]	**91.78 ± 0.14**	128.30	3.99	1173.44	**69.70 ± 0.12**	128.40	3.80	1170.40
	`Ours`	91.67 ± 0.23	**6.42**	**0.21**	**61.76**	69.40 ± 0.37	**6.42**	**0.20**	**61.60**
CMC	[44]	**92.35 ± 0.24**	128.30	3.93	1192.81	**73.56 ± 0.17**	128.40	3.92	1164.7
	`Ours`	91.65 ± 0.13	**6.42**	**0.21**	**61.17**	72.23 ± 0.24	**6.42**	**0.21**	**61.30**
DML	[80]	91.00 ± 0.04	128.30	4.01	1161.00	**72.90 ± 0.04**	128.40	3.74	1109.40
	`Ours`	**91.05 ± 0.13**	**6.42**	**0.20**	**58.05**	72.56 ± 0.15	**6.42**	**0.19**	**58.04**
		Tiny ImageNet				PASCAL VOC			
Loss	Method	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓	mAP (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓
HKD	[26]	60.46 ± 0.07	975	19.09	5595.31	65.45 ± 0.07	1371	31.25	9235.39
	`Ours`	**60.53 ± 0.31**	**48.75**	**1.04**	**304.88**	**65.48 ± 0.15**	**72.17**	**1.69**	**499.21**
CMC	[44]	**62.76 ± 0.17**	975	20.21	5949.93	**64.87 ± 0.13**	1371	32.27	9527.55
	`Ours`	59.84 ± 0.11	**48.75**	**1.00**	**293.10**	64.67 ± 0.12	**72.17**	**1.70**	**501.45**
DML	[80]	63.30 ± 0.11	975	19.38	5748.64	**65.10 ± 0.09**	1371	30.24	8903.52
	`Ours`	**63.35 ± 0.07**	**48.75**	**1.02**	**302.56**	64.98 ± 0.24	**72.17**	**1.68**	**494.64**

https://doi.org/10.1371/journal.pone.0285668.t004

when using the stochastic technique are drastically low compared to tuning for knowledge distillation. ***(2)*** The FLOPs count on every dataset is similar for its respective losses as the models used are identical for all the losses. The losses do not add to the FLOPs count of the model. The FLOPs of the models increase for larger datasets due to pixel size and instances. The results of the CIFAR 100 dataset are similar in energy consumption and carbon footprint production compared to the CIFAR 10 dataset, as both datasets have the same number of instances in their train and validation set and the same pixel size. ***(3)*** The performance measure for every dataset is a test side performance of the evaluation set of the dataset, which consists of unseen images. The efficiency metrics (FLOPs, Energy, CO_2_eq) are measured during training to estimate the training efficiency of tuning instead of using the stochastic technique. ***(4)*** The stochastic technique outperforms tuning heavily in terms of efficiency and produces similar results in comparison. Using the CMC loss on the CIFAR 10 dataset, the tuned temperature performs significantly better, achieving an accuracy of 92.35% compared to the stochastic technique at 91.65%. However, tuning the framework for the CMC loss consumes about 20 times more energy and produces CO_2 equal to a car running for 9.91km. On the CIFAR 10 dataset, the stochastic technique performs similarly to a tuned temperature. However, the stochastic approach consumes 0.20kWh of energy and produces a negligible carbon footprint of 58.05g, whereas tuning requires 4.0kWh of energy. The same trend can be seen across all the datasets. HKD loss on the Tiny Imagenet data performs almost equally for both the tuned and stochastic temperature, with a difference of 0.06% favoring the stochastic technique. Additionally, the stochastic approach consumes 1.04kWh of energy and produces 304.88g of CO_2, which is negligible compared to tuning, which consumes 19.09kWh of energy and produces 5595.31g of CO_2. The CMC loss for Tiny ImageNet performs very well compared to the stochastic technique. For the DML loss on the Tiny ImageNet dataset, the stochastic approach performs better by 0.05% against a tuned temperature. Moreover, tuning produces 19.38g of CO_2 and consumes 1.02kWh of energy which is significantly lower than tuning, which requires 19.38kWh to be tuned and makes 5748.64g of CO_2 that is comparable to driving a car for 47.8km. ***(5)*** Performance of the stochastic technique and tuned temperature are pretty

Table 5. Different performance metrics of ResNet18-MobileNetV2 as teacher-student model combination using CIFAR 10 dataset. In addition to maintaining low carbon footprints, `Ours` method performs similarly to the KD [26] method.

Model	Accuracy	Precision	Recall	Specificity	Kappa	Prevalence	F1
Teacher	94.80	0.95	0.95	0.95	0.94	0.09	0.95
KD [26]	91.67	0.92	0.92	0.92	0.90	0.09	0.92
`Ours`	91.78	0.92	0.92	0.92	0.91	0.09	0.92

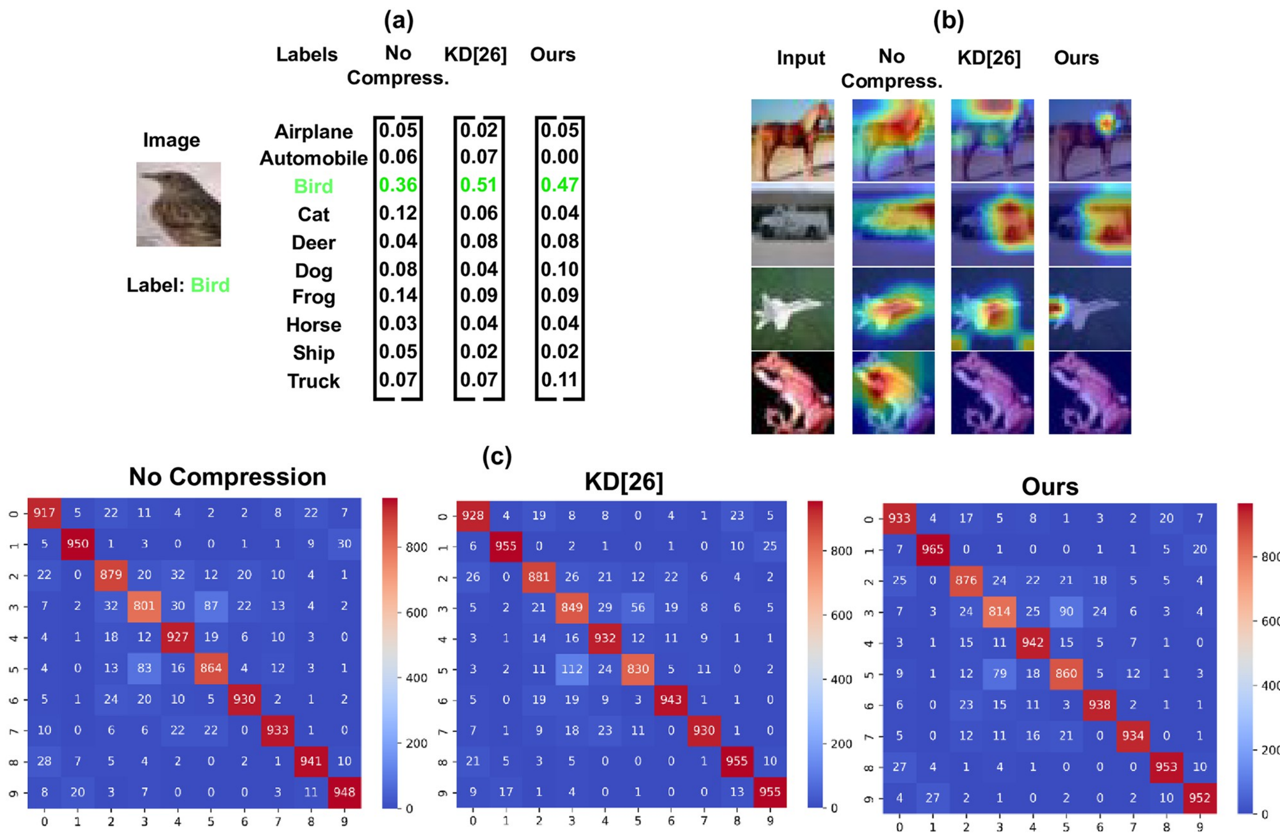
https://doi.org/10.1371/journal.pone.0285668.t005

similar across the HKD loss as the HKD loss primarily uses temperature for knowledge transfer. Quite a similar trend can also be seen for DML loss on all the datasets, as the DML loss can also be used with a temperature for knowledge transfer. For the CMC loss, performance is higher with a tuned temperature. This is because the CMC loss is generally not used with a temperature hyperparameter. It is worth noting that the CMC loss performs better with a tuned temperature than with its base form. The DML loss serves better with a tuned temperature and the stochastic technique against the base DML loss. (6) The stochastic technique contributes towards Green AI by replacing the requirement of tuning in model compression frameworks. The efficiency of the stochastic technique is considered as opposed to tuning and a commendable alternative to use.

In Table 5, we further report Precision, Recall, Kappa, Specificity, Prevalence, and F1 Score for ResNet18-MobileNetV2 teacher-student combination on the CIFAR 10 dataset. The teacher model has an accuracy of 94.8% and produces average precision and recall of 0.95, exhibiting that the models have learned all the classes properly. Similarly, the student models achieve nearly equal recall and precision (0.92), corresponding to the accuracy of 91.67% (Tuned) and 91.78% (Ours). We also compute the F1 score which is the harmonic mean of precision and recall. Here, F1 scores of the Teacher model (0.95), KD model (0.92), and Our model (0.92) effectively show that the training is not biased. The Kappa score of our model (0.91) also establishes that the performances and outputs produced are highly agreeable with the labels of the dataset.

In Fig 4, we further visualize the output of No compression, KD [26], and `Ours` methods. The output logit shown in Fig 4(a) indicates that all methods have correctly classified a sample bird image input because the class bird gets the highest score. Fig 4(b) showcases Grad-Cam images from the last layer of the MobileNetV2 architecture. `Ours` and KD [26] methods produced similar Grad-cam visualization, suggesting that both have learned from the same teacher. High values at the diagonal positions of the confusion matrices in Fig 4(c) demonstrate that `Ours` method is equally successful with No compression and KD [26] methods.

Detection. The PASCAL VOC dataset results in Table 4 (second row) show performances for object detection. For the HKD loss, our proposed stochastic technique is similar to a tuned temperature with a minor accuracy difference of 0.03%. However, traditional tuning uses 31.25 kWh of energy. Tuning such an object detection framework produces 9235.39g of CO_2, equivalent to a car running for 78.66 Km. Comparatively, our stochastic technique minimizes the CO_2 consumption by about 18 times and requires only 1.69 kWh of energy. For the CMC loss, the tuned temperature performs better (64.87% accuracy) compared to the 64.67% accuracy of the stochastic technique. However, the cost of tuning is similarly high, requiring 1.70 kWh of energy and producing 9527.55g of CO_2 as opposed to the stochastic approach. Our method requires a reduced consumption of energy and CO_2 production of 1.70kwh and 9527.55g, respectively. Similarly, the DML loss performs quite similarly in terms of performance compared to the stochastic technique. Still, the traditional KD approach is costly in energy and carbon footprint compared to the stochastic approach.



Labels	No Compress.	KD[26]	Ours
Airplane	0.05	0.02	0.05
Automobile	0.06	0.07	0.00
Bird	0.36	0.51	0.47
Cat	0.12	0.06	0.04
Deer	0.04	0.08	0.08
Dog	0.08	0.04	0.10
Frog	0.14	0.09	0.09
Horse	0.03	0.04	0.04
Ship	0.05	0.02	0.02
Truck	0.07	0.07	0.11

No Compression

	0	1	2	3	4	5	6	7	8	9
0	917	5	22	11	4	2	2	8	22	7
1	5	950	1	3	0	0	1	1	9	30
2	22	0	879	20	32	12	20	10	4	1
3	7	2	32	801	30	87	22	13	4	2
4	4	1	18	12	927	19	6	10	3	0
5	4	0	13	83	16	864	4	12	3	1
6	5	1	24	20	10	5	930	2	1	2
7	10	0	6	6	22	22	0	933	1	0
8	28	7	5	4	2	0	2	1	941	10
9	8	20	3	7	0	0	0	3	11	948

KD[26]

	0	1	2	3	4	5	6	7	8	9
0	928	4	19	8	8	0	4	1	23	5
1	6	955	0	2	1	0	1	0	10	25
2	26	0	881	26	21	12	22	6	4	2
3	5	2	21	849	29	56	19	8	6	5
4	3	1	14	16	932	12	11	9	1	1
5	3	2	11	112	24	830	5	11	0	2
6	5	0	19	19	9	3	943	1	1	0
7	7	1	9	18	23	11	0	930	1	0
8	21	5	3	5	0	0	0	1	955	10
9	9	17	1	4	0	1	0	0	13	955

Ours

	0	1	2	3	4	5	6	7	8	9
0	933	4	17	5	8	1	3	2	20	7
1	7	965	0	1	0	0	1	1	5	20
2	25	0	876	24	22	21	18	5	5	4
3	7	3	24	814	25	90	24	6	3	4
4	3	1	15	11	942	15	5	7	1	0
5	9	1	12	79	18	860	5	12	1	3
6	6	0	23	15	11	3	938	2	1	1
7	5	0	12	11	16	21	0	934	0	1
8	27	4	1	4	1	0	0	0	953	10
9	4	27	2	1	0	2	0	2	10	952

Fig 4. Visual illustration of output produced by No compression, KD [26] and `Ours` methods. (a) Output logits for a sample 'bird' image from the CIFAR 10. (b) Grad-Cam visualization of four sample images from the CIFAR 10 using the MobileNetV2. ResNet18 is used as. (c) Confusion matrices of prediction.

https://doi.org/10.1371/journal.pone.0285668.g004

Data-free knowledge distillation (DFKD) results

Knowledge distillation can be a useful model compression technique. However, it also brings about data requirement constraints, which might be unavailable due to privacy concerns [56, 58, 81]. As a result, DFKD is widely popular since it removes training data constraints by generating synthetic/artificial training images/instances [82] for many computer vision [83] and natural language processing [84] tasks. Due to the significance of DFKD, this paper also looks into the implementation and performance of DFKD frameworks. For experiments on DFKD, we use two prominent DFKD techniques: Contrastive Model Inversion (CMI) [77] and Deep Inversion (DI) [57]. The CMI looks into inverting a model to create a diverse distinguishable group of synthetic images using data diversity as an optimization objective. DI also produces artificial images by inverting learning models but optimizing inputs and regularizing feature maps using information stored in batch normalization. Table 6 represents experimental results obtained on the CMI and DI data-generation techniques on CIFAR 10 and CIFAR 100 datasets using a tuned temperature against our proposed stochastic approach. For DI, using the stochastic temperature on CIFAR 10 dataset had a similar performance to fine-tuning (83.09% compared to 83.04%). Nonetheless, fine-tuning produces a massive carbon footprint of 1369.46g of CO_2 compared to 72.07g of CO_2 produced by the stochastic technique. The stochastic process performs similarly to the tuned temperature, with a difference of 0.25% for CIFAR 10 and 0.31% for CIFAR 100. The stochastic approach produces results higher than the

Table 6. Accuracy and carbon footprints obtained on different data-free knowledge distillation (DFKD) techniques. ↑ (↓) means higher (lower) is better.

		CIFAR 10				CIFAR 10			
DFKD	**Method**	**Accuracy (%) ↑**	**GFLOPs (M) ↓**	**Energy (kWh) ↓**	**CO_2_eq (g) ↓**	**Accuracy (%) ↑**	**GFLOPs (M) ↓**	**Energy (kWh) ↓**	**CO_2_eq (g) ↓**
CMI	Tuned	**87.38**	149.18	4.56	1353.66	**57.91**	149.18	4.94	1390.8
	Ours	87.13 ± 0.21	**7.85**	**0.24**	**71.24**	57.60 ± 0.31	**7.85**	**0.26**	**73.20**
DI	Tuned	83.04	149.18	4.75	1369.46	**61.34**	149.18	4.75	1393.08
	Ours	**83.09 ± 0.14**	**7.85**	**0.25**	**72.07**	61.24 ± 0.21	**7.85**	**0.25**	**72.32**

https://doi.org/10.1371/journal.pone.0285668.t006

second most optimum temperature yet consumes 19 times less power and produces likewise less CO_2 than the tuned technique. The stochastic design optimizes hyperparameters for data-free knowledge distillation without a costly tuning process and delivers performance similar to the tuning method with lower cost, time, and carbon production.

Varying student-teacher models. To prove the robustness of our proposed method, in Table 7, we experiment with different teacher-student combinations using the Tiny ImageNet dataset. Our proposed stochastic method can work satisfactorily irrespective of strong (large models) or weak (light models) students and teachers. When using a strong teacher and a weak student for model compression, the experiment using a stochastic temperature has a 0.07% increase in accuracy compared to a tuned approach. When using self-knowledge distillation using two light models, the tuned experiments have a test accuracy of 60.32%, and the stochastic process has a test accuracy of 60.38%. Even such light model combinations can produce the same amount of CO_2 as an average car if it travels for 37.3 Km. Table 7 also showcases that with an increase in the number of model parameters for the student or teacher, there is an increase in performance. The change in FLOPs on the same dataset is due to using different student-teacher architectures. The stochastic technique produces satisfactory results and outweighs tuning in efficiency significantly.

Model compression using quantization

In addition to KD, model quantization can be another way of model compression. In Table 8, we compare a quantization method named quantization-aware-training [85] with the KD methods used in this paper. Deep learning models store parameters as floating points (32 bits or 64 bits) to achieve high precision and accuracy. However, quantization-based methods quantize the precision of input and parameters by reducing bit-width to integers (8-bit), reducing the model size, inference time, and performance. Table 8 showcases that the KD technique consumes 3.99kWh of energy and produces 1173.44g of CO_2, and performs with an

Table 7. Performance of different student-teacher combinations using Tiny ImageNet dataset. Our stochastic method consistently performs a lower carbon footprint than the tuned KD approach, keeping similar accuracy. ↑ (↓) means higher (lower) is better.

	Weak Teacher: ShuffleNetV2 (1.5M parameters)				Strong Teacher: ResNet18 (11.2M parameters)			
	Weak Student: ShuffleNetV2 (1.5M parameters)				Weak Student: ShuffleNetV2 (1.5M parameters)			
Method	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓
Tuned	60.32 ± 0.08	155.48	15.26	4489.89	60.46 ± 0.07	975.23	19.09	5595.31
Ours	**60.38 ± 0.11**	**7.77**	**0.81**	**238.41**	**60.53 ± 0.31**	**48.75**	**1.04**	**304.88**
	Weak Student: ShuffleNetV2 (1.5M parameters)				Strong Student: ResNet18 (11.2M parameters)			
	Strong Student: ResNet18 (11.2M parameters)				Strong Student: ResNet18 (11.2M parameters)			
Method	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓
Tuned	**63.66 ± 0.12**	1057	32.30	9503.34	**63.69 ± 0.06**	1877.63	34.47	10143.34
Ours	63.38 ± 0.23	**52.85**	**1.58**	**466.17**	63.45 ± 0.17	**93.83**	**1.85**	**544.94**

https://doi.org/10.1371/journal.pone.0285668.t007

Table 8. Comparison among quantization and KD methods for model compression. Experiments are done on MobileNetV2 architecture on the CIFAR 10 dataset. KD techniques use ResNet18 as the teacher. `Ours` method achieves the best performance in both accuracy and carbon footprint metrics.

Method	Accuracy (%) ↑	GFLOPs (M) ↓	Energy (kWh) ↓	CO_2_eq (g) ↓
No compression/KD	90.52	2.16	0.16	45.02
Quantization [85]	89.97	**2.46**	0.40	124.42
KD [26]	91.62	128.30	3.99	1173.44
`Ours`	**91.78**	6.42	**0.21**	**61.76**

https://doi.org/10.1371/journal.pone.0285668.t008

accuracy of 91.62%. The MobileNetV2 model produced by quantization-aware training builds a smaller and faster model but lacks performance with an accuracy of 89.97%. The quantized model consumes significantly lower energy as opposed to KD. Nonetheless, our proposed stochastic technique consumes lower power (0.21kWh), produces a smaller carbon footprint overall, and performs similarly to both methods.

Conclusion

The deployment of large models is infeasible to mobile and edge computing devices. Knowledge distillation provides a solution to this issue by generating a significantly lighter and deployable model on mobile and edge computing devices. However, we demonstrate that this suitability often comes at the expense of substantial environmental costs, largely ignored in deep learning literature. The objective of this article is threefold: (1) to investigate environmental costs for deep learning model compression using knowledge distillation, (2) to propose a stochastic approach as a means to mitigate carbon footprints for the knowledge distillation process, and (3) to conduct extensive experiments that demonstrate the suitability of the proposed stochastic approach. We propose that deep learning research should not be measured only by performance metrics such as accuracy and mAP, rather the performance metrics should also include environmental costs. Extensive experiments were conducted using various student-teacher model combinations (based on ResNet18, MobileNetV2, and ShuffleNetV2) to solve image classification and object detection problems. We estimated the carbon footprints of the overall computation process for each student-teacher model combination in terms of FLOPs count in millions, energy consumption in kWh, and CO2 equivalent in grams. Based on our empirical findings, KD consumes 13 to 18 times more carbon than the heavier teacher model. The repetitive tuning of hyperparameters is primarily responsible for such astronomical environmental costs. This article investigates the environmental impact of model compression techniques based on knowledge distillation and proposes a stochastic approach that requires less computation without sacrificing performance. Empirical results demonstrate that the proposed stochastic approach applied on CIFAR 100 dataset consumes 0.20kWh of energy and emits 61.60g of CO_2 while the tuning approach consumes 3.80kWh of energy and emits 1170.40g of CO2 with a performance difference favoring the tuning approach of 0.30%. Similarly, for the Tiny ImageNet dataset, the performance accuracy of the stochastic approach is 60.53 percent, compared to tuning's performance accuracy of 60.46 percent. However, the stochastic technique only consumes 1.04kWh of energy and emits 304.88g of CO2, whereas tuning approach emits a massive amount of carbon footprint (5,595.31g) and uses a tremendous amount of energy (19.09kWh). In future, we intend to investigate the carbon footprint of additional deep-learning tasks, such as continuous learning, domain adaptation, and meta-learning, in which the same model undergoes multiple training rounds.

Author Contributions

Conceptualization: Kazi Rafat, Shafin Rahman, Nabeel Mohammed.

Data curation: Kazi Rafat.

Formal analysis: Kazi Rafat.

Funding acquisition: Fuad Rahman.

Investigation: Kazi Rafat.

Methodology: Kazi Rafat, Sadia Islam, Abdullah Al Mahfug, Md. Ismail Hossain, Nabeel Mohammed.

Project administration: Fuad Rahman, Sifat Momen, Nabeel Mohammed.

Software: Kazi Rafat.

Supervision: Shafin Rahman, Nabeel Mohammed.

Validation: Kazi Rafat.

Visualization: Kazi Rafat, Shafin Rahman, Nabeel Mohammed.

Writing – original draft: Kazi Rafat, Sadia Islam, Abdullah Al Mahfug, Shafin Rahman, Nabeel Mohammed.

Writing – review & editing: Kazi Rafat, Sifat Momen, Shafin Rahman, Nabeel Mohammed.

References

1. LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015; 521(7553):436–444. https://doi.org/10.1038/nature14539 PMID: 26017442
2. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 7263–7271.
3. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer; 2016. p. 21–37.
4. Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:180402767. 2018;.
5. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2980–2988.
6. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Communications of the ACM. 2017; 60(6):84–90. https://doi.org/10.1145/3065386
7. Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. Citeseer. 2009;.
8. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. International journal of computer vision. 2015; 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
9. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018;.
10. Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, et al. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:190910351. 2019;.
11. Senaras C, Niazi MKK, Lozanski G, Gurcan MN. DeepFocus: detection of out-of-focus regions in whole slide digital images using deep learning. PloS one. 2018; 13(10):e0205387. https://doi.org/10.1371/journal.pone.0205387 PMID: 30359393
12. Nirschl JJ, Janowczyk A, Peyster EG, Frank R, Margulies KB, Feldman MD, et al. A deep-learning classifier identifies patients with clinical heart failure using whole-slide images of H&E tissue. PloS one. 2018; 13(4):e0192726. https://doi.org/10.1371/journal.pone.0192726 PMID: 29614076
13. Yasaka K, Abe O. Deep learning and artificial intelligence in radiology: Current applications and future directions. PLoS medicine. 2018; 15(11):e1002707. https://doi.org/10.1371/journal.pmed.1002707 PMID: 30500815

14. Yoro KO, Daramola MO. CO2 emission sources, greenhouse gases, and the global warming effect. In: Advances in carbon capture. Elsevier; 2020. p. 3–28.

15. Chauhan BS, Mahajan G, Randhawa RK, Singh H, Kang MS, et al. Global warming and its possible impact on agriculture in India. Advances in agronomy. 2014; 123:65–121. https://doi.org/10.1016/B978-0-12-420225-2.00002-9

16. Root TL, Price JT, Hall KR, Schneider SH, Rosenzweig C, Pounds JA. Fingerprints of global warming on wild animals and plants. Nature. 2003; 421(6918):57–60. https://doi.org/10.1038/nature01333 PMID: 12511952

17. DeSantis LR, Feranec RS, MacFadden BJ. Effects of global warming on ancient mammalian communities and their environments. PLoS One. 2009; 4(6):e5750. https://doi.org/10.1371/journal.pone.0005750 PMID: 19492043

18. Malcolm JR, Liu C, Neilson RP, Hansen L, Hannah L. Global warming and extinctions of endemic species from biodiversity hotspots. Conservation biology. 2006; 20(2):538–548. https://doi.org/10.1111/j.1523-1739.2006.00364.x PMID: 16903114

19. Hansen J, Sato M, Hearty P, Ruedy R, Kelley M, Masson-Delmotte V, et al. Ice melt, sea level rise and superstorms: evidence from paleoclimate data, climate modeling, and modern observations that 2 C global warming could be dangerous. Atmospheric Chemistry and Physics. 2016; 16(6):3761–3812. https://doi.org/10.5194/acp-16-3761-2016

20. Gupta E. Global warming and electricity demand in the rapidly growing city of Delhi: A semi-parametric variable coefficient approach. Energy economics. 2012; 34(5):1407–1421. https://doi.org/10.1016/j.eneco.2012.04.014

21. Van Aalst MK. The impacts of climate change on the risk of natural disasters. Disasters. 2006; 30(1):5–18. https://doi.org/10.1111/j.1467-9523.2006.00303.x PMID: 16512858

22. Khasnis AA, Nettleman MD. Global warming and infectious disease. Archives of medical research. 2005; 36(6):689–696. https://doi.org/10.1016/j.arcmed.2005.03.041 PMID: 16216650

23. Strubell E, Ganesh A, McCallum A. Energy and policy considerations for deep learning in NLP. arXiv preprint arXiv:190602243. 2019;.

24. Kim J, Jung J, Kang U. Compressing deep graph convolution network with multi-staged knowledge distillation. Plos one. 2021; 16(8):e0256187. https://doi.org/10.1371/journal.pone.0256187 PMID: 34388224

25. Deng L, Li G, Han S, Shi L, Xie Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. Proceedings of the IEEE. 2020; 108(4):485–532. https://doi.org/10.1109/JPROC.2020.2976475

26. Hinton G, Vinyals O, Dean J, et al. Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531. 2015;2(7).

27. Sadowski P, Collado J, Whiteson D, Baldi P. Deep learning, dark knowledge, and dark matter. In: NIPS 2014 Workshop on High-energy Physics and Machine Learning. PMLR; 2015. p. 81–87.

28. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. Journal of machine learning research. 2012; 13(2).

29. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

30. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 4510–4520.

31. Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 6848–6856.

32. Le Y, Yang X. Tiny imagenet visual recognition challenge. CS 231N. 2015;7(7):3.

33. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results;. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

34. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results;. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

35. Sarzynska-Wawer J, Wawer A, Pawlak A, Szymanowska J, Stefaniak I, Jarkiewicz M, et al. Detecting formal thought disorder by deep contextualized word representations. Psychiatry Research. 2021; 304:114135. https://doi.org/10.1016/j.psychres.2021.114135 PMID: 34343877

36. Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:180911096. 2018;.

37. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:171201815. 2017;.

38. Strubell E, Ganesh A, McCallum A. Energy and policy considerations for modern deep learning research. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 13693–13696.

39. Schwartz R, Dodge J, Smith NA, Etzioni O. Green ai. Communications of the ACM. 2020; 63(12):54–63. https://doi.org/10.1145/3381831

40. Parcollet T, Ravanelli M. The energy and carbon footprint of training end-to-end speech recognizers. Hal. 2021;.

41. Tamburrini G. The AI carbon footprint and responsibilities of AI scientists. Philosophies. 2022; 7(1):4. https://doi.org/10.3390/philosophies7010004

42. Anthony LFW, Kanding B, Selvan R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. arXiv preprint arXiv:200703051. 2020;.

43. Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining; 2006. p. 535–541.

44. Ba J, Caruana R. Do deep nets really need to be deep? Advances in neural information processing systems. 2014;27.

45. Liu Y, Chen K, Liu C, Qin Z, Luo Z, Wang J. Structured knowledge distillation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 2604–2613.

46. Park W, Kim D, Lu Y, Cho M. Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 3967–3976.

47. Li Z, Hoiem D. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence. 2017; 40(12):2935–2947. https://doi.org/10.1109/TPAMI.2017.2773081 PMID: 29990101

48. Liu Y, Sheng L, Shao J, Yan J, Xiang S, Pan C. Multi-label image classification via knowledge distillation from weakly-supervised detection. In: Proceedings of the 26th ACM international conference on Multimedia; 2018. p. 700–708.

49. Wang J, Gou L, Zhang W, Yang H, Shen HW. Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation. IEEE transactions on visualization and computer graphics. 2019; 25(6):2168–2180. https://doi.org/10.1109/TVCG.2019.2903943 PMID: 30892211

50. Peng B, Jin X, Liu J, Li D, Wu Y, Liu Y, et al. Correlation congruence for knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 5007–5016.

51. Chen G, Choi W, Yu X, Han T, Chandraker M. Learning efficient object detection models with knowledge distillation. Advances in neural information processing systems. 2017;30.

52. Li C, Qu X, Gnanasambandam A, Elgendy OA, Ma J, Chan SH. Photon-limited object detection using non-local feature matching and knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 3976–3987.

53. Guo Q, Wang X, Wu Y, Yu Z, Liang D, Hu X, et al. Online knowledge distillation via collaborative learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 11020–11029.

54. Dai X, Jiang Z, Wu Z, Bao Y, Wang Z, Liu S, et al. General instance distillation for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 7842–7851.

55. Chawla A, Yin H, Molchanov P, Alvarez J. Data-free knowledge distillation for object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021. p. 3289–3298.

56. Zhang Y, Chen H, Chen X, Deng Y, Xu C, Wang Y. Data-free knowledge distillation for image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 7852–7861.

57. Yin H, Molchanov P, Alvarez JM, Li Z, Mallya A, Hoiem D, et al. Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 8715–8724.

58. Choi Y, Choi J, El-Khamy M, Lee J. Data-free network quantization with adversarial knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. p. 710–711.

59. Nayak GK, Mopuri KR, Shaj V, Radhakrishnan VB, Chakraborty A. Zero-shot knowledge distillation in deep networks. In: International Conference on Machine Learning. PMLR; 2019. p. 4743–4751.

60. Micaelli P, Storkey AJ. Zero-shot knowledge transfer via adversarial belief matching. Advances in Neural Information Processing Systems. 2019;32.

61. Li H, Kadav A, Durdanovic I, Samet H, Graf HP. Pruning filters for efficient convnets. arXiv preprint arXiv:160808710. 2016;.

62. Han S, Pool J, Tran J, Dally W. Learning both weights and connections for efficient neural network. Advances in neural information processing systems. 2015;28.

63. Liu Z, Li J, Shen Z, Huang G, Yan S, Zhang C. Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2736–2744.

64. Courbariaux M, Bengio Y, David JP. Binaryconnect: Training deep neural networks with binary weights during propagations. Advances in neural information processing systems. 2015;28.

65. Courbariaux M, Hubara I, Soudry D, El-Yaniv R, Bengio Y. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:160202830. 2016;.

66. Nwankpa C, Ijomah W, Gachagan A, Marshall S. Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:181103378. 2018;.

67. De Boer PT, Kroese DP, Mannor S, Rubinstein RY. A tutorial on the cross-entropy method. Annals of operations research. 2005; 134(1):19–67. https://doi.org/10.1007/s10479-005-5724-z

68. Joyce JM. Kullback-leibler divergence. In: International encyclopedia of statistical science. Springer; 2011. p. 720–722.

69. Liashchynskyi P, Liashchynskyi P. Grid search, random search, genetic algorithm: A big comparison for NAS. arXiv preprint arXiv:191206059. 2019;.

70. Ascierto R. Uptime Institute 2019 Data Center Survey. Technical report, Uptime Institute; 2019.

71. Jafari A, Rezagholizadeh M, Sharma P, Ghodsi A. Annealing knowledge distillation. arXiv preprint arXiv:210407163. 2021;.

72. Liu J, Liu B, Li H, Liu Y. Meta Knowledge Distillation. arXiv preprint arXiv:220207940. 2022;.

73. Li Z, Li X, Yang L, Zhao B, Song R, Luo L, et al. Curriculum Temperature for Knowledge Distillation. arXiv preprint arXiv:221116231. 2022;.

74. Kobayashi T. Extractive Knowledge Distillation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2022. p. 3511–3520.

75. Huang C, Zhang Z, Mao B, Yao X. An overview of artificial intelligence ethics. IEEE Transactions on Artificial Intelligence. 2022;. https://doi.org/10.1109/TAI.2022.3194503

76. Gou J, Yu B, Maybank SJ, Tao D. Knowledge distillation: A survey. International Journal of Computer Vision. 2021; 129(6):1789–1819. https://doi.org/10.1007/s11263-021-01453-z

77. Fang G, Song J, Wang X, Shen C, Wang X, Song M. Contrastive Model Inversion for Data-Free Knowledge Distillation. arXiv preprint arXiv:210508584. 2021;.

78. Zagoruyko S, Komodakis N. Wide residual networks. arXiv preprint arXiv:160507146. 2016;.

79. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014;.

80. Zhang Y, Xiang T, Hospedales TM, Lu H. Deep mutual learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 4320–4328.

81. Nayak GK, Mopuri KR, Chakraborty A. Effectiveness of arbitrary transfer sets for data-free knowledge distillation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2021. p. 1430–1438.

82. Lopes RG, Fenu S, Starner T. Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:171007535. 2017;.

83. Chen H, Wang Y, Xu C, Yang Z, Liu C, Shi B, et al. Data-free learning of student networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 3514–3522.

84. Ma X, Shen Y, Fang G, Chen C, Jia C, Lu W. Adversarial self-supervised data-free distillation for text classification. arXiv preprint arXiv:201004883. 2020;.

85. Jacob B, Kligys S, Chen B, Zhu M, Tang M, Howard A, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 2704–2713.