Birlesik Dünya Arastırma
BD CENTER
Innovasyon ve Yayıncılık Merkezi

Global Journal of Computer Sciences: Theory and Research

Global Journal of Computer Sciences Theory and Research

Volume 10, Issue 2, (2020) 57-65 www.gjcs.eu

A new strategy for curriculum learning using model distillation

Kaan Karakose*, Bursa Uludag University, Department of Computer Engineering, Bursa, Turkey
https://orcid.org/0000-0002-4136-7592

Metin Bilgin, Bursa Uludag University, Department of Computer Engineering, Bursa, Turkey
https://orcid.org/0000-0002-4216-0542

Suggested Citation:

Karakose, K. & Bilgin, M. (2020). A new strategy for curriculum learning using model distillation. *Global Journal of Computer Sciences: Theory and Research*. *10*(2), 57-65. https://doi.org/10.18844/gjcs.v10i2.5810

Received from June 25, 2020; revised from August 14, 2020; accepted from October 22, 2020.
Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, Cyprus.

Abstract

In recent years, deep neural networks have been successful in both industry and academia, especially for computer vision tasks. Humans and animals learn much better when gradually presented in a meaningful order showing more concepts and complex samples rather than randomly presenting the information. The use of such training strategies in the context of artificial neural networks is called curriculum learning. In this study, a strategy was developed for curriculum learning. Using the CIFAR-10 and CIFAR-100 training sets, the last few layers of the pre-trained on ImageNet Xception model were trained to keep the training set knowledge in the model's weight. Finally, a much smaller model was trained with the sample sorting methods presented using these difficulty levels. The findings obtained in this study show that the accuracy value generated when trained by the method we provided with the accuracy value trained with randomly mixed data was more than 1% for each epoch.

Keywords: Curriculum learning, model distillation, deep learning, academia, neural networks.

* ADDRESS FOR CORRESPONDENCE: Kaan Karakose, Bursa Uludag University, Department of Computer Engineering, Bursa, Turkey. *E-mail address*: kaankarakose@uludag.edu.tr / Tel.: +90-224-275-5289

1. Introduction

Humans and animals learn much better when examples are gradually presented in a meaningful order that shows more concepts and complex ones rather than randomly presenting the information. Implementing such training strategies in artificial neural networks is called 'curriculum learning'. When the data set examples are organised according to a curriculum, the model may find a better local minimum point [1]. The question asked by the researchers in previous studies [2]–[4] was 'Can machine learning algorithms benefit from a similar training strategy?'

Researchers have scrutinised this question over time. The study [1], which provides an answer to this question, intuitively created a curriculum in deep artificial neural networks for the first time.

The examples in the data set used in this study were presented intuitively in one order. The data set used consisted of shapes. The researchers considered the shapes to be challenging for humans, such as rhombus and parallelogram. The samples later presented to the model were square, rectangular and triangular shapes, which were presented to the network in the first stage. With this ordering, the model found a better local minimum point and learned better.

Another study [5], which relies on shaping the educational examples in the education stage without prior ordering, is the learning method introduced as self-paced learning. In this strategy, a warming phase expressed as 'warm-up' is realised by training a few epochs with the network training samples. After this process ends, the weights are updated according to the loss that will occur in the forward transition of the sample. In other words, a 'lambda' is defined, which will determine its effect on weights. In each epoch, the model's loss function is shaped according to the loss of this lambda value and the incoming sample and the process continues.

Making educational examples into a curriculum and the performance of the model were discussed [6]–[10]. Although most of the methods used are heuristic approaches, theoretical approaches [11], [12] are essential for this area. In this study, studies were conducted on CIFAR-10 [13] and CIFAR-100 [13] datasets. These data sets were chosen for reasons such as having different comprehensible examples, being composed of enough classes, a small sample size, and low calculation costs. The grading of training samples was studied using the model distillation method [14]. It aimed to create a curriculum using the information obtained by distilling the information of the data set from the teacher model layers. In the present study, educational samples were placed in different orders statically, and their effect on model success was examined. Methods were compared with each other and the good results were discussed.

2. Related work

After demonstrating [1] that giving examples to the learner in a meaningful order provides an optimisation, various methods have been proposed to increase the applicability of such strategies. Kumar et al. [5] proposed the self-paced learning method to organise training examples at the education stage without any prior order. Goodfellow et al. [7] made use of the distance of each sample from each other and from the centre point parameter of the clustering algorithm used in unsupervised learning while determining the difficulty levels. Gou et al.[15] used the density distribution of the data in the feature space to investigate the sample difficulties. The authors described samples that were close to the mean value of this distribution as 'clean'. They specified the categories they determined according to this density distribution as 'clean', 'noisy' and 'high noisy', respectively.

In the object recognition problem, the difficulties of Wang et al. [10] were determined by the number of objects found in a sample belonging to the data set. The authors categorised 'difficult' and 'easy' according to the number of objects found in the sample. In the multi-task classification study for visual features [16], the sample difficulties were determined by the correlation they created among themselves. According to the correlation value of an example, it was separated as 'strongly' and 'weakly'. The samples were presented to the network in this order. [17] proposed three methods for the difficulties of the examples in their study on text data set. In brief, the first method is that examples to the network with increasing sentence lengths after the first epoch. The second method is that presenting the examples according to average sentence lengths to the network. The last method presents examples to the network, starting from the examples with the shortest sentence length. [6] suggested using sentence lengths and underused words in sentences as difficulty levels, similar to Han and Myaeng's study [17]. Examples that contain the longer sentence or are composed of words belonging to a set of less used words are categorised as 'difficult' and presented to the network according to this order. This study is a study on neural machine translation. The method used is an intuitive approach.

Curriculum learning has been positively influencing the success of the model, making improvements in models using different strategies, using smaller models effectively, finding a better local minimum for optimisation algorithm and giving high success in different artificial neural network model topologies.

3. Methods and materials

3.1. Model distillation and teacher model

Model distillation transfers knowledge from a large model to a smaller model without loss of validity. Since smaller models are cheaper to evaluate, they can be placed on less powerful hardware [14]. In the distillation of knowledge, types of knowledge, distillation strategies and teacher–student architectures play a crucial role in student learning. In this study, the response-based knowledge [18] distillation strategy, which authors categorised and named, was used. In the present study, it is foreseen to use the information in layers to create a curriculum from a large model trained for a data set. In Figure 1, the diagram of the information distillation process is shown. Generally, the last layer is used when extracting knowledge from the teacher model. The size of this layer varies according to the number of classes. A short training process was applied to the teacher model. After the training process, the logits of each sample were softened or hardened using a temperature Softmax function.

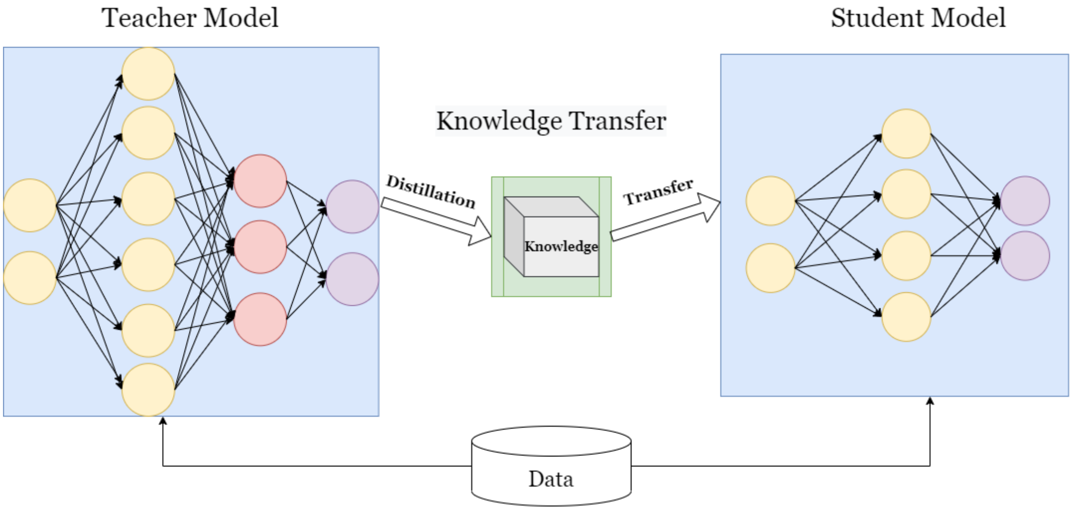

Figure 1. Knowledge distillation [18]

In this study, the pre-trained on ImageNet Xception [19] model was trained as a teacher model. The transfer learning [20] approach was applied for the training teacher model. Only the last layers were trained and the other layers were frozen. Thus, the basic information of the data was kept in the last layers that trained on the data set. Table 1 includes the result of the teacher model training.

Table 1. Teacher model parameters for CIFAR-10 and CIFAR-100 data sets

Data sets	Trainable parameters	Non-trainable Parameters	Train accuracy	Test accuracy	Loss	Epoch
CIFAR-100	11,147,876	16,110,632	0.9421	0.3744	1.2936	100
CIFAR-10	11,147,876	16,110,632	0.9833	0.6603	0.201	80

As shown in Table 1, the teacher model trained on CIFAR-100 was with a higher number of epochs than the CIFAR-10 data set. In addition, while the last layer of the teacher model varied according to the number of classes, the penultimate layer was initialised according to the size of the sample belonging to the data set. In experimental studies, the training parameters of this model were manipulated to find a better model accuracy. The best epoch was 100 for the CIFAR-100 and 80 for the CIFAR-10 data set. Once the data set was stored in the model layers, the teacher model was ready to identify the example's difficulties.

3.2. Sample's difficulties and the student model

In this subsection, the curriculum for the student model was produced from extracting the information from the layers of the teacher model, which was previously trained for ImageNet, and then warmed up for the data sets we used. To do this process, two different definitions were made: label loss and sample loss. The definition of teacher model layers is shown in Figure 2. Here, $L^{(1)}$ refers to the last layer and $L^{(2)}$ refers to the penultimate layer.

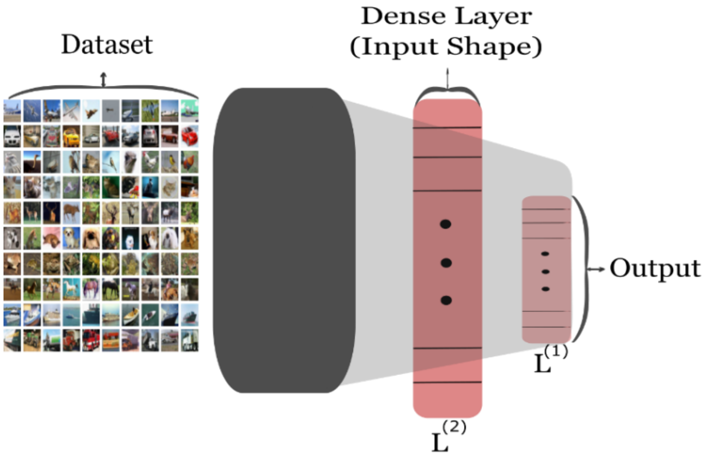

Figure 2. The teacher's model layer

The following methods were used to calculate the label loss. First, each sample was presented to the teacher network. The teacher model produced different results for each layer output. Using the information distillation method [14] the cross-entropy given in Equation (1) was calculated between the label and the output of the final layer of the teacher model. In this equation, N_s represents the class number, y_i represents the label vector and $L_i^{(1)}$ represents the layer output vector of the related sample. The optimal temperature parameter T for the temperature Softmax function used in the model distillation method was found during experimental studies, which is 0.6.

$$\text{Label loss} = -\frac{1}{N_s}\sum_{i=1}^{N_s}\left[y_i \log \text{softmax}(L_i^{(1)}, T) + (1 - y_i\ ) \log(1 - \text{softmax}(L_i^{(1)}, T))\right] \quad (1)$$

The unit number of the dense layer was $L^{(2)}$ and the same was chosen as the data shaped to calculate the sample loss. In other words, since the shape of the data of CIFAR-10 and CIFAR-100 are 32 × 32, the number of units in the $L^{(2)}$ layer was initialised as 1,024. After that, the correlation between the pixel values of the data and $L^{(2)}$layer was calculated. The sample loss was calculated as the cross-correlation shown in Equation (2). This operation will have a value between 1 and -1.

$$\text{Sample loss} = \frac{\sum_m^M \sum_n^N (X_{m,n} - \overline{X})(L^{(2)}_{m,n} - \overline{L^{(2)}})}{\sqrt{\left(\sum_n^N (X_{m,n} - \overline{X})^2\right)\left(\sum_n^N \left(L^{(2)}_{m,n} - \overline{L^{(2)}}\right)^2\right)}}.$$
$$\overline{X} = \frac{1}{M+N}\sum\nolimits_m \sum\nolimits_n X_{m,n} \qquad (2)$$
$$\overline{L^{(2)}} = \frac{1}{M+N}\sum\nolimits_m \sum\nolimits_n L^{(2)}_{m,n}$$

Sample loss and label loss values were different for each sample. For each example, the sum of the values as shown in Equation (3) was defined as the 'difficulty' score of the sample in this study. These difficulties ordered different ways to create a curriculum.

$$\text{Difficulty} = \text{Label loss} + \text{Sample loss} \qquad (3)$$

In this study, the difficulties ordered four different sorting methods. These were descending, ascending, class-based descending (CBD) and class-based ascending (CBA). The first two methods were the standard ordering of data according to their difficulty score. For class-based ordering, samples belonging to each class were first sorted within themselves and then sorted according to the suitable batch size for the model and the operating environment. This sorting method is shown in Figure 3.

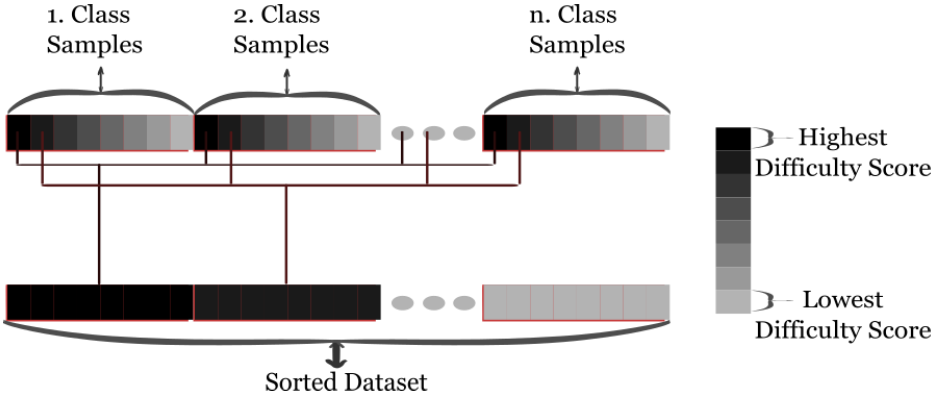

Figure 3. Class-based sorting

Here, the basic idea is creating the batch with equal numbers from the hardest or easiest examples of each class. This approach is more consistent with the heuristic methods of curriculum learning.

When data sets were ordered as mentioned above, in other words, when the curriculum was created, the student model could be trained according to these curricula. A vanilla convolutional neural network with far fewer parameters and layers than the teacher model was used as the student model. The student model used is shown in Figure 4.

CIFAR-10 and CIFAR-100 data sets were arranged according to the sorting methods used and mixed sorting situations called 'shuffled'. The student model was trained on these ordering.

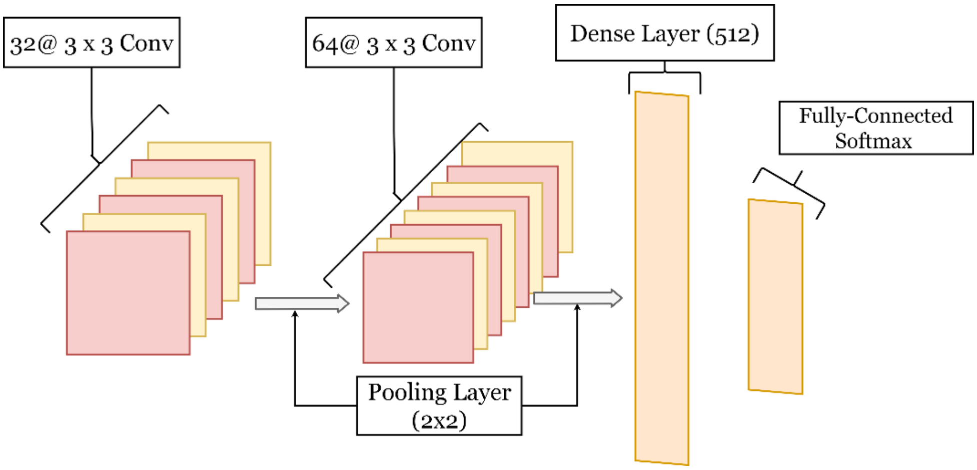

Figure 4. The student model

4. Results

Since the student model was small, it was trained on seven epochs. The reason for training seven epochs was that the model was so small, and the saturation was reached quickly when the number of epochs increased. However, the main idea was to show the effects of curriculum learning on neural networks. To better analyse the performance of the student model, the model trained with each sorting method was trained 10 times, and the accuracies per epoch were averaged. As shown in Figure 5, the presented CBA and CBD sorting methods increased the model's accuracy by producing better results in each epoch compared to the shuffled case.

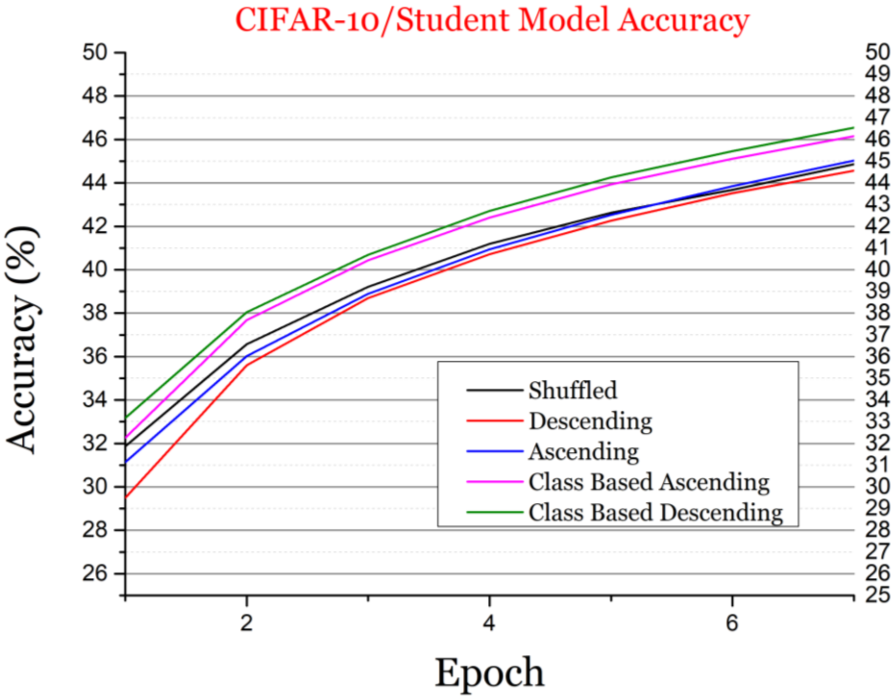

Figure 5. The student model accuracy on CIFAR-10 for different sorting

The sorting case of descending and vice versa is worse than the mixed case because the data belonging to the same class in the sorting are often seen in the same batch. Consequently, the student model will form its weights according to the samples belonging to a single class that coincides with each batch. As a result, it will be challenging to find a general minimum point. This situation will continue to affect the overall success of the system negatively. Using CBA and CBD sorting methods offered to get rid of this situation; this problem disappeared and the model's success increased.

As shown in Figure 6, the student model accuracy for CIFAR-100 resulted in similar situation as for CIFAR-10.

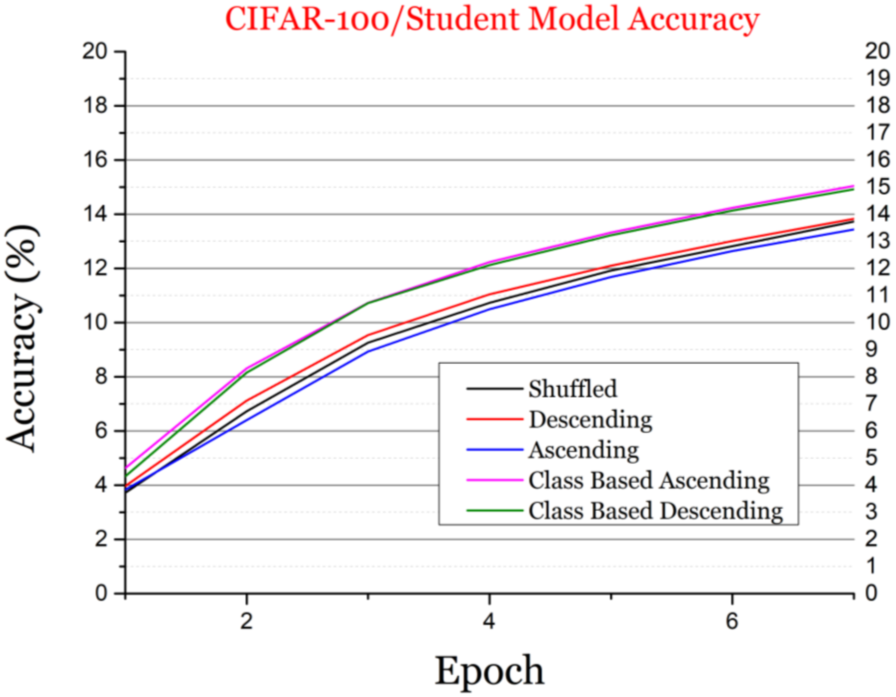

Figure 6. The student model accuracy on CIFAR-100 for different sorting

The difference between CBA and CBD and shuffled sustained was kept throughout the training. The difference that occurred here was that the ascending sorting method and vice versa, which were below and above the values of the shuffled case. That the model was too small for this data set brought about this poor situation. However, the hypothesis defended in this study is that the different ordering of the training samples will affect the model's success. Another difference was that the winning sorting methods produce results that were very close to each other.

Table 2 shows the cycle-based differences between the CIFAR-100 and CIFAR-10 data set results and the mixed case. The CBA and CBD methods we presented in both data sets resulted in 1%–1.5% better for each epoch.

Table 2. Differences between the present methods and shuffled case for both data sets based on their accuracy

	CIFAR-10		CIFAR-100	
Epoch	CBA	CBD	CBA	CBD
1.	0.400%	1.312%	0.909%	0.612%
2.	1.115%	1.486%	1.579%	1.421%
3.	1.231%	1.472%	1.474%	1.458%
4.	1.201%	1.510%	1.508%	1.393%
5.	1.307%	1.630%	1.403%	1.307%
6.	1.435%	1.787%	1.413%	1.316%
7.	1.293%	1.691%	1.322%	1.195%

5. Conclusion and discussion

In this study, it was observed how the model accuracy was changed by just adjusting the places of the samples in the data set with a specific order. The success of the presented methods in CIFAR-10 and CIFAR-100 data sets was discussed. The difficulties of data created by extracting the information about the data set from the model are sorted, and how it affects the model accuracy has been shown.

The main idea in this study is that when the places of the examples are changed according to a specific rule, the model may increase the learning accuracy. The findings obtained in this study showed that the accuracy value generated when trained by the method we provided with the accuracy value trained with randomly mixed data was more than 1% for each epoch.

It can be said that the reason why the sample difficulties are directly sorting methods are worse than the randomly mixed case because the data belonging to the same class correspond to more than one repetition in the sorting. It has been possible to increase the model's success with the CBA and CBD methods for both data sets. With these methods, artificial neural network applications can be made more effective in small devices such as microprocessors and mobile phones with less processing power. In addition, investigating the effects of sample loss defined for each sample can be a starting point for further research.

References

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1--8, doi: 10.1145/1553374.1553380.

[2] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71--99, Jul. 1993, doi: 10.1016/0010-0277(93)90058-4.

[3] D. L. T. Rohde and D. C. Plaut, "Language acquisition in the absence of explicit negative evidence: How important is starting small?" *Cognition*, vol. 72, no. 1, pp. 67--109, 1999, doi: 10.1016/S0010-0277(99)00031-1.

[4] K. A. Krueger and P. Dayan, "Flexible shaping: How learning in small steps helps," *Cognition*, vol. 110, no. 3, pp. 380--394, 2009, doi: 10.1016/j.cognition.2008.11.014.

[5] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010. [Online]. Available: https://proceedings.neurips.cc/paper/2010/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf

[6] E. A. Platanios, O. Stretcu, G. Neubig, B. Poczos, and T. M. Mitchell, "Competence-based curriculum learning for neural machine translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL HLT)*, vol. 1, 2019, pp. 1162--1172, doi: 10.18653/v1/n19-1119.

[7] I. J. Goodfellow \emph{et al.}, "Wolfram research, 'FER-2013' from the wolfram data repository," Wolfram Res., Champaign, IL, USA, Tech. Rep., 2018. [Online]. Available: https://datarepository.wolframcloud.com/resources/FER-2013

[8] T. Yamashita and T. Watasue, "Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 853--857, doi: 10.1109/ICIP.2014.7025171.

[9] W. Zaremba and I. Sutskever, "Learning to execute," pp. 1--25, 2014, \emph{arXiv:1410.4615}. [Online]. Available: http://arxiv.org/abs/1410.4615

[10] J. Wang, X. Wang, and W. Liu, "Weakly- and semi-supervised faster R-CNN with curriculum learning," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2018, pp. 2416--2421, doi: 10.1109/ICPR.2018.8546088.

[11] D. Weinshall and D. Amir, "Theory of curriculum learning, with convex loss functions," *J. Mach. Learn. Res.*, vol. 21, pp. 1--18, Jan. 2020. [Online]. Available: http://arxiv.org/abs/1812.03472

[12] D. Weinshall, G. Cohen, and D. Amir, “Curriculum learning by transfer learning: Theory and experiments with deep networks,” in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 12, 2018, pp. 8331--8339. [Online]. Available: https://arxiv.org/abs/1802.03796

[13] A. Krizhevsky. (2009). *Learning Multiple Layers of Features From Tiny Images*. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[14] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” pp. 1--9, 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[15] S. Guo \emph{et al.}, “CurriculumNet: Weakly supervised learning from large-scale web images,” in *Computer Vision* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11214. 2018, pp. 139--154, doi: 10.1007/978-3-030-01249-6_9.

[16] N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. A. Kakadiaris, “Curriculum learning for multi-task classification of visual attributes,” in *Proc. IEEE Int. Conf. Comput. Vis. Work. (ICCVW)*, Jan. 2018, pp. 2608--2615, doi: 10.1109/ICCVW.2017.306.

[17] S. Han and S. H. Myaeng, “Tree-structured curriculum learning based on semantic similarity of text,” in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 971--976, doi: 10.1109/ICMLA.2017.00-27.

[18] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” pp. 1--36, 2020, \emph{arXiv:2006.05525}. [Online]. Available: https://arxiv.org/abs/2006.05525, doi: 10.1007/s11263-021-01453-z.

[19] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2017, pp. 1800--1807, doi: 10.1109/CVPR.2017.195.

[20] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345--1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.