Curriculum Learning with Diversity for Supervised Computer Vision Tasks

Anonymous Authors[1]

Abstract

Curriculum learning techniques are a viable solution for improving the accuracy of automatic models, by replacing the traditional random training with an easy to hard strategy. However, the standard curriculum methodology does not automatically provide improved results, but is constrained by multiple elements like the data distribution or the proposed model. In this paper, we introduce a novel curriculum sampling strategy which takes into consideration the diversity of the training data together with the difficulty of the inputs. We determine the difficulty using a state-of-the-art difficulty estimator and we model the diversity during training, giving higher priority to elements from classes visited less. We conduct object detection and instance segmentation experiments on Pascal VOC 2007 and Cityscapes data sets, surpassing both the randomly-trained baseline and the standard curriculum approach. We prove that our strategy is very efficient in unbalanced data sets, leading to faster convergence and more accurate results, where other curriculum-based strategies fail.

1. Introduction

Although the quality of automatic models highly increased with the development of deep and very deep neural networks, an important and less studied key element for the overall performance is the training strategy. In this direction, (Bengio et al., 2009) introduced curriculum learning (CL), a set of learning strategies inspired by the way humans teach and learn. People learn the easiest concepts at first, followed by more and more complex elements. Similarly, CL proposes feeding the automatic model with easier samples at the beginning of the training, while gradually introducing more difficult data as the training proceeds.

The idea is straightforward, but an important question is how to determine whether a sample is easy or hard. CL requires the existence of a predefined metric which can compute the difficulty of the input examples. In this paper, we use the image difficulty estimator from (Ionescu et al., 2016) which is based on the time required by human annotators to identify if a class is present or not in a certain image.

The next challenge is building the curriculum schedule, or the rate at which we can augment the training set with more complex information. To address this problem, we follow a sampling strategy similar to the one introduced in (Soviany et al., 2020). Based on the difficulty score, we sample according to a probability function, which favors easier samples in the first iterations, but converges to give the same weight to all the examples in the later phases of the training. Still, the probability of sampling a harder example in the first iterations is not null, and the more difficult samples which are occasionally picked increase the diversity of the data and help training.

The above-mentioned methodology should work well for balanced data sets, but it can fail when the data is unbalanced. (Ionescu et al., 2016) show that some classes may be more difficult than others. This can make our sampling strategy neglect examples from certain classes and slow down training. A more serious problem arises when the data is biased towards the easier classes. To solve this problem, we add a new term to our sampling function which takes into consideration the classes of the elements already sampled, in order to give more importance to images from less-visited classes.

Since it is a sampling procedure, our CL approach can be applied to any supervised task in machine learning. In this paper, we focus on object detection and instance segmentation, two of the main tasks in computer vision, which require the model to identify the class and the location of objects in images. To test the validity of our approach, we experiment on two data sets: Pascal VOC 2007 (Everingham et al., a) and Cityscapes (Cordts et al., 2016), and compare our curriculum with diversity strategy against the standard random training method, a curriculum sampling (without diversity) procedure and an inverse-curriculum approach, which selects images from hard to easy. We employ a state-of-the-art Faster R-CNN (Ren et al., 2015) detector with a Resnet-101 (He et al., 2016) backbone for the object detection experiments, and a Mask R-CNN (He et al., 2017) model based on Resnet-50 for instance segmentation.

Our main contributions can be resumed as follows:

1. We illustrate the necessity of adding diversity when using CL in unbalanced data sets;
2. We introduce a novel curricular sampling function, which takes into consideration the class-diversity of the training samples and improves results when traditional curriculum approaches fail;
3. We prove our strategy by experimenting on two computer vision tasks: object detection and instance segmentation, using two data sets of high interest.

We organize the rest of this paper as follows. In Section 2, we present the most relevant related works and compare them with our approach. In Section 3, we explain in detail the methodology we follow. We present our results in Section 4, and draw our conclusion and discuss possible future work in the last section.

2. Related Work

Curriculum learning. (Bengio et al., 2009) introduced the idea of curriculum learning (CL) to train artificial intelligence, proving that the standard learning paradigm used in human educational systems could also be applied to automatic models. CL represents a class of easy-to-hard approaches, which have successfully been employed in a wide range of machine learning applications, from natural language processing (Guo et al., 2019; Kocmi & Bojar, 2017; Liu et al., 2018; Platanios et al., 2019; Subramanian et al., 2017), to computer vision (Gong et al., 2016; Gui et al., 2017; Hacohen & Weinshall, 2019; Jiang et al., 2018; Li et al., 2017; Shi & Ferrari, 2016; Weinshall & Cohen, 2018), or audio processing (Amodei, 2016; Ranjan & Hansen, 2017).

One of the main limitations of CL is that it assumes the existence of a predefined metric which can rank the samples from easy to hard. These metrics are usually task-dependent with various solutions being proposed for each. For example, in text processing, the length of the sentence can be used to estimate the difficulty of the input (shorter sentences are easier) (Platanios et al., 2019; Spitkovsky et al., 2009), while the number and the size of objects in a certain sample can provide enough insights about difficulty in image processing tasks (images with few large objects are easier) (Shi & Ferrari, 2016; Soviany & Ionescu, 2018). In our paper, we employ the image difficulty estimator of Ionescu et al. (Ionescu et al., 2016) which was trained considering the time required by human annotators to identify the presence of certain classes in images.

To alleviate the challenge of finding a predefined difficulty metric, Kumar et al. (Kumar et al., 2010) introduce Self-paced learning (SPL), a set of approaches in which the model ranks the samples from easy to hard during training, based on its current status. For example, the inputs with the smaller losses at a certain time during training are easier than the samples with higher losses. Many papers apply SPL successfully (Sangineto et al., 2018; Supancic & Ramanan, 2013; Tang et al., 2012), and some methods combine prior knowledge with live training information, creating self-paced with curriculum techniques (Jiang et al., 2015; Zhang et al., 2019). Even so, SPL still has some limitations, requiring a methodology on how to select the samples and on how much to emphasize easier examples. Our approach is on the borderline between CL and SPL, but we consider it to be pure curricular, although we use training information to advantage classes visited less. The reason behind our statement is that the class of the training samples is a priori information and a similar system could iteratively select examples from every class. Still, we prefer to use the class-diversity as a term in our difficulty-based sampling probability function, in order to not massively alter the actual class distribution of our data set.

The easy-to-hard idea behind CL can be implemented in multiple ways. One option is to start training on the easiest set of images, while gradually adding more difficult batches (Bengio et al., 2009; Gui et al., 2017; Kocmi & Bojar, 2017; Shi & Ferrari, 2016; Spitkovsky et al., 2009; Zhang et al., 2018). Although most of the models keep the visited examples in the training set, in (Kocmi & Bojar, 2017) the authors reduce the size of each bin until combining it with the following one, in order to use each example only once during an epoch. In (Liu et al., 2018; Soviany et al., 2020) the authors propose a sampling strategy according to some probability function, which favors easier examples in the first iterations. As the authors show, the easiness score from (Soviany et al., 2020) could also be added as a new term to the loss function to emphasize the easier examples in the beginning of the training. In this paper, we follow their sampling strategy and use a similar probability function to select training examples.

Despite leading to good results in many related papers, the standard CL procedure is highly influenced by the task and the data distribution. Simple tasks may not gain much from using curriculum approaches, while employing CL in unbalanced data sets can lead to slower convergence. To address the second problem, (Jiang et al., 2014; Sachan & Xing, 2016) argue that a key element is diversity. (Jiang et al., 2014) introduce a SPL with diversity technique in which they regularize the model using both difficulty information and the variety of the samples. They suggest using clustering algorithms to split the data in diverse groups. (Sachan & Xing, 2016) measure diversity using the angle between the hyperplanes the samples induce in the feature space. They chose the examples that optimize a convex combination of the curriculum learning objective and the sum of angles

between the candidate samples and the examples selected in previous steps. In our model, we define diversity based on the classes of our data. We combine our predefined difficulty metric with a score which favors images from classes visited less, in order to sample easy and diverse examples at the beginning of the training, then gradually add more complex elements. Our idea works well for supervised tasks, but it can be extended to unsupervised learning by replacing the ground-truth labels with a clustering model, like suggested in (Jiang et al., 2014).

Object detection is the task of predicting the location and the class of objects in certain images. As noted in (Soviany & Ionescu, 2018), the state-of-the-art object detectors can be split in two main categories: two-stage and single stage models. The two-stage object detectors (He et al., 2017; Ren et al., 2015) use a Region Proposal Network to generate regions of interest which are then fed to another network for object localization and classification. The single stage approaches (Liu et al., 2016; Redmon et al., 2016) take the whole image as input and solve the problem like a regular regression task. These methods are usually faster, but less accurate than the two-stage designs. **Instance segmentation** is similar to object detection, but more complex, requiring to generate a mask instead of a bounding box for the objects in the test image. Our strategy can be applied on any detection and segmentation model, but, in order to increase the relevance of our results, we experiment with high quality Faster R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017) baselines.

3. Methodology

Training artificial intelligence using curriculum approaches, from easy to hard, can lead to improved results in a wide range of tasks (Amodei, 2016; Gong et al., 2016; Gui et al., 2017; Guo et al., 2019; Hacohen & Weinshall, 2019; Jiang et al., 2018; Kocmi & Bojar, 2017; Li et al., 2017; Liu et al., 2018; Platanios et al., 2019; Ranjan & Hansen, 2017; Shi & Ferrari, 2016; Subramanian et al., 2017; Weinshall & Cohen, 2018). Still, it is not simple to determine which samples are easy or hard, the available metrics usually being task-dependent. Another challenge of CL is finding the right curricular schedule, i.e. how fast to add more difficult examples to training, as well as introducing the right quantity of harder samples at the right time to positively influence convergence. In this section, we present our approach for estimating difficulty and our curriculum sampling strategy which we enhance by taking into consideration the diversity of the examples.

3.1. Difficulty estimation

In (Ionescu et al., 2016), the authors defined image difficulty as the required human time for solving a visual search task. They collected annotations for the Pascal VOC 2012 (Everingham et al., b) data set, which they normalized and fed as training data for their regression model. We follow their strategy as described in the original paper (Ionescu et al., 2016) to determine the difficulty scores of the images in our data sets. These scores have values ≈ 3, with a larger score defining a more difficult sample. We translate the values between $[-1, 1]$ using Equation 1 to simplify the usage of the score in the next steps.

$$Scale_{min-max}(x) = \frac{2 \cdot (x - min(x))}{max(x) - min(x)} - 1 \quad (1)$$

3.2. Curriculum sampling

(Soviany et al., 2020) introduces a curriculum sampling strategy, which favors easier examples in the first iterations and converges as the training progresses. It has the advantage of being a continuous method, removing the necessity of a curricular schedule for enhancing the difficulty-based batches. Furthermore, the fact that it is a probabilistic sampling method does not constrain the model to only select easy examples in the first iterations, as batching does, but adds more diversity in data selection. We follow their approach in building our curriculum sampling strategy with only a small change to better emphasize the difficulty. We use the following function to assign weights to the input images during training:

$$w(x_i, t) = (1 - diff(x_i) \cdot e^{-\gamma \cdot t})^k, \forall x_i \in X, \quad (2)$$

where x_i is the training example from the data set X, t is the current iteration, and $diff(x_i)$ is the difficulty score associated with the selected sample. γ is a parameter which sets how fast the function converges to 1, while k sets how much to emphasize the easier examples. Our function varies from the one proposed in (Soviany et al., 2020) by changing the position of the k parameter. We consider that we can take advantage of the properties of the power function which increases faster for numbers greater than the unit. Since $1 - s_i \cdot e^{-\gamma \cdot t} \in [0, 2]$, and the result is > 1 for easier examples, our function will focus more on the easier samples in the first iterations. We transform the weights into probabilities and we sample according to them.

3.3. Curriculum with diversity sampling

As (Jiang et al., 2014; Sachan & Xing, 2016) note, applying a CL strategy does not guarantee improved quality, the diversity of the selected samples having a great impact on the final results. A simple example is the case in which the data set is biased, having fewer samples of certain classes. Since some classes are more difficult than others (Ionescu et al., 2016), if the data set is not well-balanced, the model will not visit the harder classes until the later stages of the training.

This is the reason why we enhance our sampling method, by adding a new term, which is based on the diversity of the examples.

Our diversity scoring algorithm is simple, taking into consideration the classes of the selected samples. During training, we count the number of visited objects from each class ($num_{objects}(c)$). We subtract the mean the values to determine how often was each class visited. This is formally presented in Equation 3. We scale and translate the results between $[-1, 1]$ using Equation 1 to get the score of each class, then, for every image, we compute the image-level diversity by averaging the class score for each object in its ground-truth labels (Equation 4).

$$visited(c_i) = num_{objects}(c_i) - \frac{\sum_{c_j \in C} num_{objects}(c_j)}{|C|} \quad \forall c_i \in C. \quad (3)$$

$$imgVisited(x_i) = \frac{\sum_{obj \in objects(x_i)} visited(class(obj))}{|objects(x_i)|} \quad \forall x_i \in X. \quad (4)$$

In our diversity algorithm we want to emphasize the images with objects from classes visited less, i.e. with a small $imgVisited$ value, closer to -1. We compute a scoring function similar to Equation 2, which also takes into consideration how often a class was visited, in order to add diversity:

$$w(x_i, t) = [1 - \alpha \cdot (diff(x_i) \cdot e^{-\gamma \cdot t}) - (1 - \alpha) \cdot (imgVisited(x_i) \cdot e^{-\gamma \cdot t})]^k, \quad (5)$$

where α controls the impact of each component, the difficulty and the diversity, while the rest of the notation follows Equation 2. We transform the weights into probabilities by dividing them by their sum, and we sample according to them.

4. Experiments

4.1. Data sets

In order to test the validity of our method, we experiment on two data sets: Pascal VOC 2007 (Everingham et al., a) and Cityscapes (Cordts et al., 2016). We conduct detection experiments on 20 classes, training on the 5011 images from the Pascal VOC 2007 trainval split. We perform evaluation on the test split which contains 4952 images. For our instance segmentation experiments, we use the Cityscapes data set which contains eight labeled object classes. We train on the training set of 2975 images and we evaluate on the validation split of 500 images.

4.2. Baselines and configuration

We build our method on top of the Faster R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017) implementations available at: https://github.com/facebookresearch/maskrcnn-benchmark. For our detection experiments, we use Faster R-CNN with Resnet-101 (He et al., 2016) backbone, while for segmentation we employ the Resnet-50 backbone on the Mask R-CNN model. We use the configurations available on the web site, with the learning rate adjusted for a training with a batch size of 4. In our sampling procedure (Equation 5) we set $\alpha = 0.5$, $\gamma = 6 \cdot 10^{-5}$, and $k = 5$. We do not compare with other models, because the goal of our paper is not surpassing the state of the art, but improving the quality of our baseline model. We also present the results of a hard-to-easy sampling, in order to prove the efficiency of the curriculum approach.

4.3. Evaluation metrics

We evaluate our results using the mean Average Precision (AP). The AP score is given by the area under the precision-recall curve for the detected objects. The exact evaluation protocol has some differences for each data set, thus we use the Pascal VOC 2007 (Everingham et al., a) metric for the detection experiments and the Cityscapes (Cordts et al., 2016) metric for the instance segmentation results. We use the evaluation code available at https://github.com/facebookresearch/maskrcnn-benchmark. More details about the evaluation metrics can be found in the original papers (Cordts et al., 2016; Everingham et al., a).

4.4. Results and discussion

The class distribution of the objects in Pascal VOC 2007 clearly favors class *person*, with 4690 instances, while classes *dinningtable* and *bus* only contain 215 and 229 instances, respectively. This would not be a problem if the difficulty of the classes was similar, because we can assume the test data set has a matching distribution, but this is not the case.

Figure 1 presents how the two sampling methods behave during training on the Pascal VOC 2007 data set. In the first 10k iterations, curriculum sampling selects images with almost 20k objects from class *person* and only 283 instances from class *diningtable*. By adding diversity, we lower the gap between classes, reaching 10k objects of persons and 1000 instances of tables. This behaviour continues as the training progresses, with the differences between classes

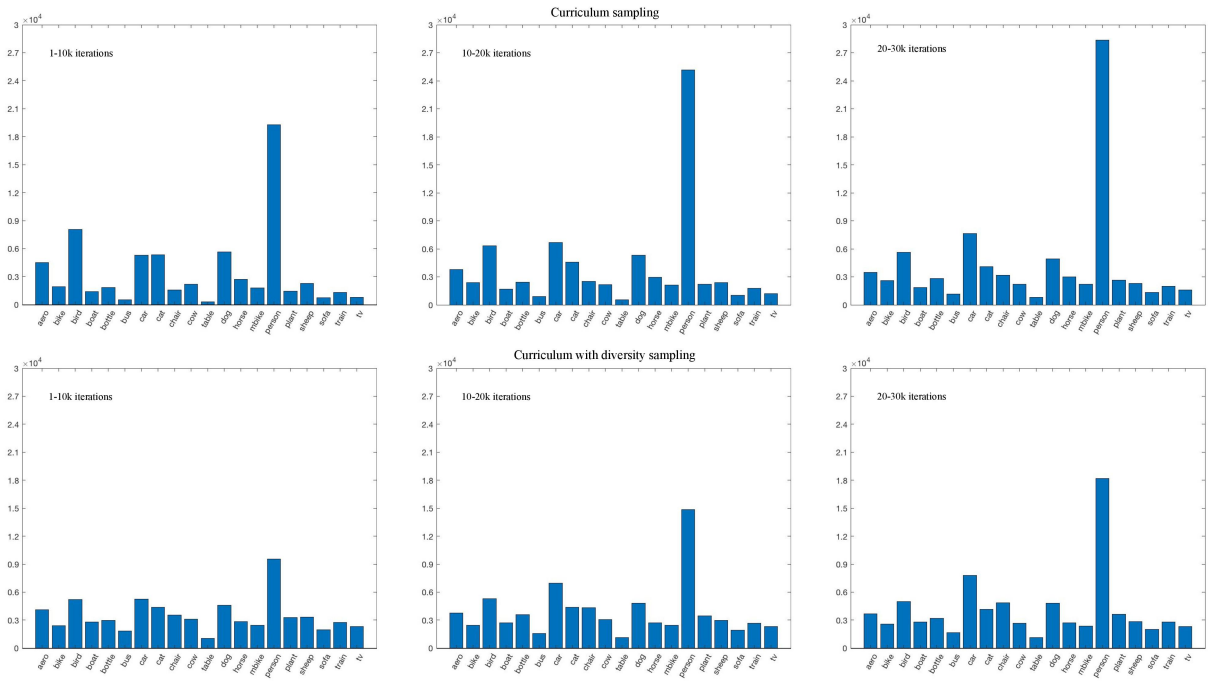

Figure 1. Number of objects from each class sampled during our training on Pascal VOC 2007. On the first row is the curriculum sampling method and on the second row is the curriculum with diversity approach. We present the first 30000 iterations for each case, with histograms generated from 10k in 10k steps.

being smaller when adding diversity. It is important to note that we do not want to sample the exact number of objects from each class, but to keep the class distribution of the actual data set, while feeding the model with enough details about every class. Figure 2 shows the difficulty of the examples sampled according to our strategies. We observe that by adding diversity we do not break our curriculum learning schedule, the examples still being selected from easy to hard.

To further prove the efficiency of our method, we compute the AP on both object detection and instance segmentation tasks. The results are presented in Tables 1 and 2.

In our object detection experiments, the sampling with diversity approach provides an improvement of 0.52% over the standard curriculum method, and of 0.76% over the randomly-trained baseline. Our experiments, with an inverse curriculum approach, from hard to easy, lead to the worst results, showing the utility of CL. Moreover, Figure 3 illustrates the evolution of the AP during training. The curriculum with diversity approach has superior results over the baseline from the beginning to the end of the training. The standard CL method, on the other hand, starts from lower scores, exactly because it does not visit enough samples from more difficult classes in the early stages of the training.

Table 1. Average Precision scores for object detection on Pascal VOC 2007 data set.

MODEL	AP
FASTER R-CNN (BASELINE)	72.24
CURRICULUM SAMPLING	72.48
INVERSE CURRICULUM SAMPLING	70.89
DIVERSE CURRICULUM SAMPLING	**73.00**

Table 2. Average Precision scores for instance segmentation on Cityscapes data set.

MODEL	AP	AP50	AP75
FASTER R-CNN (BASELINE)	38.72	69.15	34.95
CURRICULUM SAMPLING	38.47	**69.88**	35.01
INVERSE CURRICULUM	37.40	68.17	34.22
DIVERSE CURRICULUM	**39.12**	69.86	**35.4**

The instance segmentation results on the Cityscapes data set confirm the conclusion from our previous experiments. As Table 2 shows, the curriculum with diversity is again the optimal method, surpassing the baseline with 0.4% using AP, 0.71% using AP50%, and 0.45% using AP75%. It is interesting to point that, although the diverse curriculum approach has a better AP and AP75% than the standard CL method, the former technique surpasses our method with

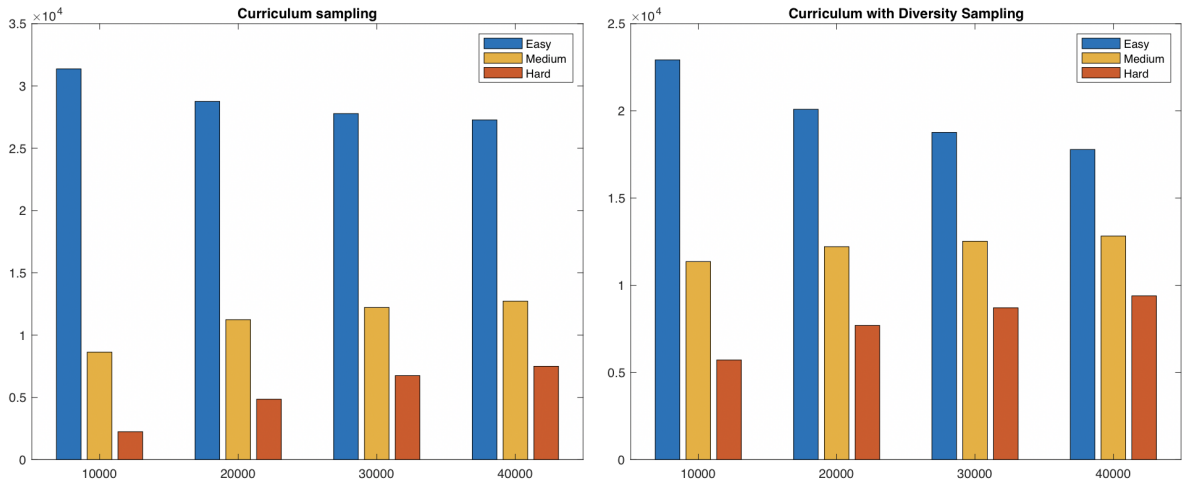

Figure 2. Difficulty of the images samples during our training on Pascal VOC 2007. On the left is presented the curriculum sampling method and on the right the curriculum with diversity approach. We present the first 40000 iterations for each case, with histograms generated from 10k in 10k steps. Best viewed in color.

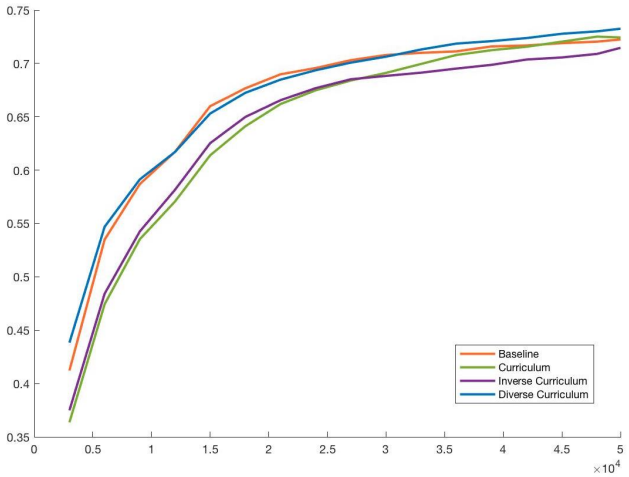

Figure 3. Evolution of mAP during training on Pascal VOC 2007 for object detection. Best viewed in color.

0.02% when evaluated using AP50%. The inverse curriculum approach has the worst scores again, strengthening our statements on the utility of curriculum learning.

5. Conclusion and future work

In this paper, we presented a simple method of optimizing the curriculum learning approaches on unbalanced data sets. We consider that the diversity of the selected examples is just as important as their difficulty, and neglecting this fact may slow down training for more difficult classes. We introduced a novel sampling function, which uses the classes of the visited examples together with a difficulty score to ensure the curriculum schedule and the diversity of the selection. Our object detection and instance segmentation experiments conducted on two data sets of high interest prove the superiority of our method over the randomly-trained baseline and over the standard CL approach. A benefit of our methodology is that it can be used on top of any deep learning model, for any supervised task. For the future work, we plan on extending the framework to unsupervised tasks, by introducing a novel method of computing the diversity of the examples.

References

Amodei, D. e. a. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proceedings of ICML*, pp. 173–182, 2016.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of ICML*, pp. 41–48, 2009.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of CVPR*, 2016.

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

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-

network.org/challenges/VOC/voc2012/workshop/index.html, b.

Gong, C., Tao, D., Maybank, S. J., Liu, W., Kang, G., and Yang, J. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260, 2016.

Gui, L., Baltrušaitis, T., and Morency, L. Curriculum learning for facial expression recognition. In *Proceedings of FG*, pp. 505–511, 2017.

Guo, J., Tan, X., Xu, L., Qin, T., Chen, E., and Liu, T.-Y. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:1911.08717*, 2019.

Hacohen, G. and Weinshall, D. On the power of curriculum learning in training deep networks. In *Proceedings of ICML*, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of CVPR*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of ICCV*, pp. 2961–2969, 2017.

Ionescu, R., Alexe, B., Leordeanu, M., Popescu, M., Papadopoulos, D. P., and Ferrari, V. How hard can it be? estimating the difficulty of visual search in an image. In *Proceedings of CVPR*, pp. 2157–2166, 2016.

Jiang, L., Meng, D., Yu, S.-I., Lan, Z., Shan, S., and Hauptmann, A. Self-paced learning with diversity. In *Proceedings of NIPS*, pp. 2078–2086, 2014.

Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. G. Self-paced curriculum learning. In *Proceedings of AAAI*, 2015.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of ICML*, pp. 2304–2313, 2018.

Kocmi, T. and Bojar, O. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of RANLP*, pp. 379–386, 2017.

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Proceedings of NIPS*, pp. 1189–1197, 2010.

Li, S., Zhu, X., Huang, Q., Xu, H., and Kuo, C. J. Multiple instance curriculum learning for weakly supervised object detection. In *Proceedings of BMVC*. BMVA Press, 2017.

Liu, C., He, S., Liu, K., and Zhao, J. Curriculum learning for natural answer generation. In *Proceedings of IJCAI*, pp. 4223–4229, 2018.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *Proceedings of ECCV*, pp. 21–37. Springer, 2016.

Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., and Mitchell, T. Competence-based curriculum learning for neural machine translation. In *Proceedings of NAACL*, pp. 1162–1172, 2019.

Ranjan, S. and Hansen, J. H. Curriculum learning based approaches for noise robust speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):197–210, 2017.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of CVPR*, pp. 779–788, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of NIPS*, pp. 91–99, 2015.

Sachan, M. and Xing, E. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of ACL*, pp. 453–463, 2016.

Sangineto, E., Nabi, M., Culibrk, D., and Sebe, N. Self paced deep learning for weakly supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):712–725, 2018.

Shi, M. and Ferrari, V. Weakly supervised object localization using size estimates. In *Proceedings of ECCV*, pp. 105–121. Springer, 2016.

Soviany, P. and Ionescu, R. T. Frustratingly Easy Trade-off Optimization between Single-Stage and Two-Stage Deep Object Detectors. In *Proceedings of CEFRL Workshop of ECCV*, pp. 366–378, 2018.

Soviany, P., Ardei, C., Ionescu, R. T., and Leordeanu, M. Image difficulty curriculum for generative adversarial networks (cugan). In *Proceedings of WACV*, 2020.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. Baby Steps: How "Less is More" in unsupervised dependency parsing. In *Proceedings of NIPS Workshop on Grammar Induction, Representation of Language and Language Learning*, 2009.

Subramanian, S., Rajeswar, S., Dutil, F., Pal, C., and Courville, A. Adversarial generation of natural language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 241–251, 2017.

Supancic, J. S. and Ramanan, D. Self-paced learning for long-term tracking. In *Proceedings of CVPR*, pp. 2379–2386, 2013.

Tang, K., Ramanathan, V., Fei-Fei, L., and Koller, D. Shifting weights: Adapting object detectors from image to video. In *Proceedings of NIPS*, pp. 638–646, 2012.

Weinshall, D. and Cohen, G. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *Proceedings of ICML*, 2018.

Zhang, D., Han, J., Zhao, L., and Meng, D. Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework. *International Journal of Computer Vision*, 127(4):363–380, 2019.

Zhang, X., Kumar, G., Khayrallah, H., Murray, K., Gwinnup, J., Martindale, M. J., McNamee, P., Duh, K., and Carpuat, M. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*, 2018.