

CBM: Curriculum by Masking

Andrei Jarcă^{a,1}, Florinel-Alin Croitoru^{a,1} and Radu Tudor Ionescu^{a,*}

^aDepartment of Computer Science, University of Bucharest, 14 Academiei, Bucharest, Romania

Abstract. We propose Curriculum by Masking (CBM), a novel state-of-the-art curriculum learning strategy that effectively creates an easy-to-hard training schedule via patch (token) masking, offering significant accuracy improvements over the conventional training regime and previous curriculum learning (CL) methods. CBM leverages gradient magnitudes to prioritize the masking of salient image regions via a novel masking algorithm and a novel masking block. Our approach enables controlling sample difficulty via the patch masking ratio, generating an effective easy-to-hard curriculum by gradually introducing harder samples as training progresses. CBM operates with two easily configurable parameters, *i.e.* the number of patches and the curriculum schedule, making it a versatile curriculum learning approach for object recognition and detection. We conduct experiments with various neural architectures, ranging from convolutional networks to vision transformers, on five benchmark data sets (CIFAR-10, CIFAR-100, ImageNet, Food-101 and PASCAL VOC), to compare CBM with conventional as well as curriculum-based training regimes. Our results reveal the superiority of our strategy compared with the state-of-the-art curriculum learning regimes. We also observe improvements in transfer learning contexts, where CBM surpasses previous work by considerable margins in terms of accuracy. We release our code for free non-commercial use at <https://github.com/CroitoruAlin/CBM>.

1 Introduction

Humans learn by grasping the easier concepts before gradually moving to the more complex ones. Inspired by this observation, Bengio *et al.* [3] proposed curriculum learning, a training regime that introduces a structured order of the data samples to train neural models from easy to hard. This method ensures that the examples are presented to the model in a logical and meaningful sequence, allowing the model to effectively learn and develop its knowledge base. The method has been employed in multiple scenarios yielding significant performance improvements [44]. Furthermore, over the course of time, various methods have been developed to integrate curriculum learning [3, 7, 9, 18, 24, 28, 34, 35, 37, 41, 42, 43]. While many of these approaches involve different components of the training process, Soviany *et al.* [44] have classified them into four main categories. The first category, known as data-level curriculum, involves sorting the data samples based on certain difficulty criteria. This approach aligns with the initial implementation of Bengio *et al.* [3]. The second category, referred to as model-level curriculum, includes methods that gradually increase the capacity of the model

as the training progresses. The third category, represented by task-level curriculum, aims to make the learning task more intricate over time. Lastly, objective-level curriculum begins with a simplified objective, *e.g.* a convex objective, and transforms it during training until it becomes the final target objective, which is usually non-convex.

The data-level curriculum learning approaches involve sorting the training examples based on some difficulty metric [3, 22, 28, 36, 43, 46, 51, 57], before proceeding with the actual learning process. However, this approach has a significant challenge associated with it, namely the need to use a custom difficulty metric for each domain. The choice of the difficulty metric may vary depending on the specific learning task, and, in certain cases, it can be very hard to propose a useful difficulty metric [9]. Building on the success of self-supervised approaches [21, 10] used to train deep learning models to reconstruct masked information (tokens), we propose a novel data-level curriculum learning approach, termed Curriculum by Masking (CBM), that artificially raises the difficulty level of each training image by masking a certain number of patches, where the number of masked patches gradually increases during the training process, as illustrated in Figure 1. Hence, our approach does not require the prior sorting of data samples, as it enables full control over the complexity of a data sample via the masking ratio. However, we conjecture that randomly masking patches does not necessarily induce an easy-to-hard curriculum. For example, if the masked patches happen to hide most of the background in an image, leaving the foreground object that needs to be recognized or detected visible, we will actually end up with an easier example instead of a more difficult one. To this end, we propose to sample the masked patches from a probability distribution derived from image gradient magnitudes, essentially prioritizing the masking of salient image patches. Since salient regions are more likely to contain discriminative patterns, masking patches with larger gradient magnitudes reduces the number of visible discriminative patterns, thus increasing the difficulty of recognizing objects in images.

We conduct experiments to determine the effectiveness of our approach in object recognition and object detection. We apply CBM on different types of convolutional and transformer neural networks, *e.g.* ResNet-18 [20], Wide-ResNet-50 [55], CvT-13 [52], and YOLOv5 [26]. Our empirical study is carried out on five data sets: CIFAR-10 [29], CIFAR-100 [29], ImageNet [40], Food-101 [4] and PASCAL VOC [13]. We compare our approach with four state-of-the-art curriculum learning methods [9, 11, 42, 50], as well as two more baseline training regimes. The first baseline is the vanilla training regime, while the second one employs the vanilla patch masking technique proposed by He *et al.* [21]. Lastly, we present ablation results and a comprehensive comparison of masking ratio schedules, identifying multiple successful configurations and parameter choices

* Corresponding Author. Email: raducu.ionescu@gmail.com.

¹ Equal contribution.

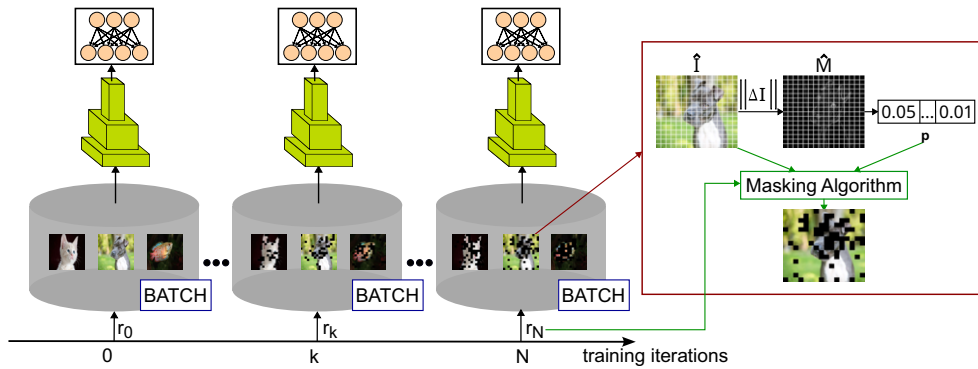


Figure 1: An overview of Curriculum by Masking. The training starts with fully visible images. During training, the patch masking ratio is gradually increased to make the samples more difficult. The masking is predominantly focused on the more salient regions (with higher gradient magnitudes), to reduce the likelihood of producing easier images by masking the background information. Best viewed in color.

for our method.

In summary, our contribution is threefold:

- We propose a novel curriculum learning method based on masking an increasingly higher number of patches during training.
- We introduce a patch selection strategy that prioritizes the masking of patches with larger gradient magnitudes, ensuring an easy-to-hard curriculum.
- We empirically demonstrate the effectiveness of CBM in object recognition and object detection for multiple neural architectures, comparing it with several competing training regimes.

2 Related Work

Curriculum learning is a training technique introduced by Bengio *et al.* [3], which provides the training examples in a meaningful order, from easy to hard, to neural networks. The objective is to enhance the performance of neural models, while also improving the convergence speed of the training process. Since its introduction, curriculum learning has proven its effectiveness in various domains, such as computer vision [3, 7, 9, 18, 24, 41, 42, 43], natural language processing [3, 9, 28, 34, 37, 45], and signal processing [1, 9, 38]. The method has been very successful and has undergone extensive development, as illustrated in some recent surveys [44, 49]. These developments range from strategies for measuring data difficulty [3, 22, 28, 36, 41, 43, 46, 51, 57] to methods focusing on other aspects of the training process [5, 9, 6, 27, 42]. A well-known method to apply curriculum learning is by defining a metric that evaluates the complexity of the data, and subsequently arranging the training examples from the simplest to the most challenging ones based on the respective metric. Researchers have made significant strides in finding improved metrics for various domains and tasks. For instance, images containing fewer and larger objects in computer vision are deemed easier than other images [41, 43]. In natural language processing, word frequency [3, 34] and sequence length [8, 28, 46, 57] are utilized to assess the sample difficulty. In some cases, researchers have also integrated human feedback into their metric design [25, 36, 51].

The aforementioned curriculum strategies have proven to be effective. However, they have been found to lack practicality due to their reliance on human expert input [25], which may not always be available. Moreover, these methods remain fixed during training and may not adapt the curriculum to the changing needs of the models. As a result, the research community developed new curriculum-based approaches to overcome these limitations. For in-

stance, Kumar *et al.* [30] introduced self-paced learning, a method that measures the difficulty of the training samples based on the performance of the trained model. Thus, the order of the training samples changes according to the model feedback during training, and thanks to this property, several works adopted the approach [14, 17, 23, 30, 31, 39, 58]. Moreover, it is possible to implement a combination of self-paced learning and classic curriculum learning approaches. This approach has been previously utilized under the name of self-paced curriculum learning [23, 44]. Another popular method is teacher-student curriculum learning [19, 53, 56], where the teacher learns to supervise the student network via a curriculum.

Methods that fall under the model-level curriculum learning paradigm [5, 9, 27, 42, 44] are closer to our work. In this setting, the curriculum does not imply ordering the samples in ascending order of their difficulty. Instead, the curriculum is implemented by increasing the model capacity as the training progresses, or by adjusting the task to become more accessible at the beginning of the training. Curriculum by Smoothing (CBS), a technique developed by Sinha *et al.* [42], blurs the activation maps resulting from convolutional layers during the training process to let the model focus on the bigger picture. CBS gradually reduces the amount of blur as the model improves. This approach has been successful on various data sets and models. However, it does require extra processing steps during training, which can make the learning process last longer. The work of Burduja *et al.* [5] is an alternative to CBS, where the input images are blurred instead of the intermediary activation maps. Another example is Learning Rate Curriculum (LeRaC) [9]. This method assigns different learning rates to the network layers based on their proximity to the input. Layers closer to the input have higher learning rates, while those farther away have lower rates. Over time, the learning rates are adjusted to converge to a consistent value. Similar to [5, 9, 42], our approach does not require an external difficulty measure.

Different from other curriculum learning approaches, we propose to induce a curriculum via a progressive masking of input patches. To the best of our knowledge, we are the first to introduce a curriculum learning method based on patch masking. Furthermore, we go beyond a naive implementation and apply the masking operation by taking into consideration the saliency (gradient magnitude) of the patches to create the premises for an easy-to-hard curriculum. This differs significantly from other approaches [5, 42] which apply the smoothing operation uniformly in space, using a single filter applied at every location of the input via convolution. Another advantage in favor of CBS [42] is that our approach does not include auxiliary operations after each neural layer of the model, so the training time is identical to that of the conventional training regime.

Algorithm 1 Saliency-Based Masking Algorithm

Input: r_k – the percentage of masked patches at iteration k ;
 \hat{I} – the input image;
 n – the number of patches in an image;
 \mathbf{p} – an array of size n containing the masking probabilities for each patch.
Output: \hat{I}_{mask} – the masked image.
Computation:

- 1: $n_{mask} \leftarrow n \cdot r_k$
- 2: $\hat{I}_{mask} \leftarrow \hat{I}$
- 3: **for** i in $1 \dots n_{mask}$ **do**
- 4: $j \sim \text{Categorical}(n, \mathbf{p})$
- 5: $\hat{I}_{mask}[j] \leftarrow 0$
- 6: **end for**
- 7: **return** \hat{I}_{mask}

3 Method

Masking specific parts, *e.g.* patches or tokens, of input data samples has been demonstrated to be a successful technique to train deep learning models in a self-supervised manner, in both natural language processing [10, 32] and computer vision [2, 15, 16, 21, 32, 47] domains. This method has been primarily used as a pre-training task [10, 21], in which the model is tasked with reconstructing the masked information. Through this approach, the model is able to learn and recognize patterns in the data, leading to improved accuracy and more effective performance in downstream tasks. In this work, we redesign the masking procedure to create a curriculum learning method based on masking patches according to their saliency level. Our approach is able to artificially generate examples of various difficulty levels. We name the resulting learning procedure Curriculum by Masking (CBM).

Our curriculum learning procedure starts from the original samples and gradually creates more difficult images as the training progresses. We increase the image difficulty by masking a higher number of heterogeneous patches. We estimate the heterogeneity (or saliency) of a patch via the average gradient magnitude computed on both horizontal and vertical axes. Such patches are likely to contain regions of interest, such as object parts or other discriminative patterns. Before the training starts, our approach creates a curriculum schedule vector $\mathbf{r} \in \mathbb{R}^N$, where each element r_k represents the percentage of patches that are to be masked during the k -th training iteration. We note that the maximum number of iterations is denoted by N , and r_N is equal to a maximum masking ratio, which is fixed beforehand through validation. We empirically study various alternative functions to generate the curriculum schedule \mathbf{r} and regulate its growth rate. For details about the results obtained with different curriculum schedules, please refer to Section 4.

Our saliency-based masking procedure is described in Algorithm 1. The algorithm operates on the input image \hat{I} , which is divided into non-overlapping patches and represented as a tensor of dimensions $\mathbb{R}^{n \times \hat{h} \times \hat{w} \times c}$, where n denotes the number of patches, \hat{h} and \hat{w} refer to the height and width of each patch, and c represents the number of color channels. In addition to the input image, the algorithm also relies on the ratio of masked patches r_k for the current training iteration k , and an array of probabilities $\mathbf{p} \in [0, 1]^n$ that controls the likelihood of masking each patch.

First, the algorithm computes the number of patches to be masked, denoted as n_{mask} , based on the specified percentage r_k . Then, at each loop iteration i , the algorithm samples a patch index j from

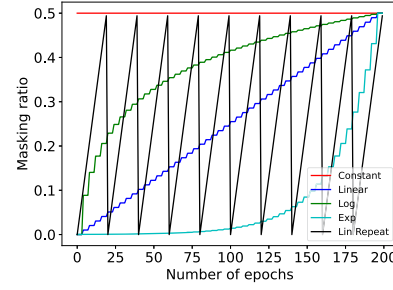


Figure 2: The proposed curriculum schedules are based on masking a certain number of image patches in each epoch. For illustration purposes, the number of epochs is $N = 200$ and the maximum masking ratio is $r_N = 0.5$ for all schedules. Best viewed in color.

the set $\{1, \dots, n\}$ according to a categorical distribution described by the vector \mathbf{p} . Then, the patch at index j is masked by setting its pixels to zero in the output image \hat{I}_{mask} .

As previously stated, the masking algorithm prioritizes salient regions of the input image. This property is accomplished by controlling the probabilities in \mathbf{p} , such that salient patches (likely to contain discriminative patterns) are assigned with higher probabilities, thus increasing their chances of being selected for masking. We hypothesize that salient regions can be identified to a certain extent by analyzing the magnitudes of the image gradients. We transform color images to the grayscale, before computing the gradients. Formally, given a grayscale image $I \in \mathbb{R}^{h \times w}$, where $h = n_h \cdot \hat{h}$, $w = n_w \cdot \hat{w}$ and $n = n_h \cdot n_w$, we compute its gradient as follows:

$$\Delta I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]. \quad (1)$$

Then, we computed the magnitude $M \in \mathbb{R}^{h \times w}$ of the gradient as follows:

$$M = \|\Delta I\| = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}. \quad (2)$$

Next, we split M into n patches and obtain the tensor denoted by $\hat{M} \in \mathbb{R}^{n \times \hat{h} \times \hat{w}}$. Finally, we compute the elements of \mathbf{p} via the following equations:

$$m_i = \frac{\sum_{j=1}^{\hat{h}} \sum_{l=1}^{\hat{w}} \hat{M}[i, j, l]}{\hat{h} \cdot \hat{w}}, \forall i \in \{1, \dots, n\}, \quad (3)$$

$$p_i = \frac{m_i}{\sum_{j=1}^n m_j}, i \in \{1, \dots, n\}, \quad (4)$$

where m_i is the unnormalized gradient magnitude of the i -th patch, and p_i is the probability of masking the i -th patch.

We underline that CBM has two important hyperparameters, namely the vector \mathbf{r} and the number of patches n . Given that the images are divided into non-overlapping patches, the number of patches n is directly determined by the patch dimensions. In our experiments, we use square patches (having the same height and width). To generate \mathbf{r} , we consider one of the various schedules depicted in Figure 2. The *constant* schedule emulates the framework proposed by He *et al.* [21], which does not represent an actual curriculum, since it generates equally difficult examples during training, *i.e.*:

$$r_k = r_N, \forall k \in \{1, 2, \dots, N\}, \quad (5)$$

where N is the number of training epochs. The *constant* schedule is only added as a baseline.

We propose three curriculum schedules that generate a *logarithmic*, *linear* or *exponential* growth of the masking ratio. The log schedule creates a more aggressive curriculum, increasing the mask-

ing ratio much sooner:

$$r_k = r_N \cdot \log_2 \left(1 + \frac{k}{N} \right), \forall k \in \{1, 2, \dots, N\}. \quad (6)$$

In contrast, the exp schedule is the least aggressive curriculum, introducing high masking ratios only at the very end:

$$r_k = r_N \cdot e^{k-N}, \forall k \in \{1, 2, \dots, N\}. \quad (7)$$

The linear schedule is between the log and exp schedule, its formula being given by:

$$r_k = r_N \cdot \frac{k}{N}, \forall k \in \{1, 2, \dots, N\}. \quad (8)$$

While humans learn from easy to hard, reminding easy concepts when trying to learn complex ones is generally helpful. To this end, we consider an additional curriculum learning schedule called *linear repeat*, which repeats the easy-to-hard curriculum multiple times during training. This schedule is depicted in Figure 2 along with the other schedules. It is repeated 10 times, at every 20 epochs, just for illustration purposes. On easy (unmasked) images, the model can learn some generic patterns from the data. As the images get harder by increasing the masking ratio, the model will try to identify new patterns that are robust to the masking procedure. In this process, the model can forget the initially learned patterns due to *catastrophic forgetting*. The linear repeat scheduler helps to avoid the catastrophic forgetting of the generic patterns learned on easy images, by reintroducing easy images at various stages during the entire optimization process.

For all the aforementioned schedules, the maximum masking ratio $0 < r_N < 1$ is a hyperparameter that can be established via grid or random search on the validation set. The rate of masking can be used to control the trade-off between underfitting and overfitting. If there is no masking, the classification task becomes easy and the model can overfit the training data. In contrast, if the masking ratio is too high, the classification task can become too hard, and the model will be unable to learn from the masked data, leading to underfitting.

One of the curriculum schedules needs to be chosen before training begins. In the experiments, we present extensive results with the proposed schedules, which generate more or less aggressive curriculum learning strategies.

4 Experiments

4.1 Experimental Setup

Data sets. We evaluate our curriculum learning method on four object recognition data sets (CIFAR-10 [29], CIFAR-100 [29], ImageNet [40] and Food-101 [4]) and one object detection data set (PASCAL VOC [13]). Each of the CIFAR-10 and CIFAR-100 data sets consists of 50,000 training images and 10,000 test images with a resolution of 32×32 pixels. CIFAR-10 contains 10 object categories, while CIFAR-100 contains 100 categories. ImageNet-1K [40] is the most popular benchmark in computer vision. We use 200 ImageNet categories in the evaluation. Food-101 [4] contains 75,750 training images and 25,250 test images from 101 food categories. PASCAL VOC 2007+2012 [13] is a well-known data set for object detection, which consists of 21,503 images. Objects from 20 categories are annotated with bounding boxes.

Architectures. To evaluate the generalization capabilities of CBM across different architectures, we perform object recognition experiments with two convolutional neural networks and one transformer model. More precisely, we employ ResNet-18 [20], Wide-ResNet-50 [55] and CvT-13 [52]. Being a transformer-based architecture, CvT benefits significantly from pre-training. Thus, we use a checkpoint

Table 1: Optimal hyperparameter configurations of the ResNet-18, Wide-ResNet-50 and CvT-13 models. For the conventional (vanilla) training regime, we report the employed optimizer, number of epochs (N), learning rate (α) and mini-batch size. In addition, for the CBM regime, we report the maximum patch masking ratio (r_N), the number of patches per image ($n_h \times n_w$), and the curriculum schedule. Note that the vector \mathbf{r} is directly determined by r_N and the schedule.

Regime	Model	ResNet-18, Wide-ResNet-50		CvT-13
	Data set	CIFAR-10 CIFAR-100 Food-101	ImageNet	All
Vanilla	optimizer	SGD	SGD	AdaMax
	weight decay	$10^{-4}/5 \cdot 10^{-4}$	10^{-4}	-
	N	200	100	40
	α	10^{-1}	10^{-1}	$5 \cdot 10^{-4}$
	mini-batch	64	64	64
CBM	r_N	0.4/0.6	0.4/0.6	0.4/0.6
	schedule	Lin repeat	Lin repeat	Lin repeat
	$n_h \times n_w$	4×4	4×4	4×4

pre-trained on ImageNet-21K in our experiments. As such, we apply CBM in both “training from scratch” and fine-tuning scenarios. Furthermore, we employ an object detection pipeline, YOLOv5 [26], to evaluate the benefits of CBM in object detection. We specifically choose the YOLOv5s [26] model, which is pre-trained on the MS COCO data set [33].

Baselines. We compare CBM with the conventional training regime, which uses the optimal hyperparameters (learning rate, batch size, weight decay, and so on) specific to each of the subsequent experiments. Moreover, we compare our curriculum learning approach with four competing curriculum learning methods, namely Curriculum by Smoothing (CBS) [42], Label-Similarity Curriculum Learning (LSCL) [11], Learning Rate Curriculum (LeRaC) [9] and EfficientTrain [50]. In the ablation study, we also compare with the framework of He *et al.* [21], which can be seen as an ablated version of CBM.

Hyperparameter tuning. The optimal hyperparameter configurations of the ResNet-18, Wide-ResNet-50 and CvT-13 models are shown in Table 1. Regardless of the training regime, ResNet-18 and Wide-ResNet-50 are trained with SGD for 200 epochs on the CIFAR-10, CIFAR-100 and Food-101 data sets, and 100 epochs on ImageNet. Since CvT-13 is pre-trained on ImageNet, it only requires 40 epochs of fine-tuning to converge. The SGD optimizer is used alongside an annealing learning rate scheduler for ResNet-18 and Wide-ResNet-50, requiring a large initial learning rate of 10^{-1} . In contrast, AdaMax does not require a scheduler, as it adjusts its learning rate based on data characteristics. We optimize all models with the cross-entropy loss. The maximum masking ratio for each model is found through grid search on the validation set, considering values between 0.1 and 0.8. The optimal maximum masking ratio for CIFAR-10 is 0.4, while ImageNet seems to benefit from a higher ratio of 0.6 when using ResNet-18 or CvT-13. Among the proposed curriculum schedules, we select the masking ratio at each epoch based on a linear schedule, which is repeated every 5 epochs. We present results with various curriculum schedules later. The number of patches is chosen between 2×2 and 16×16 . The optimal choice for all models is 4×4 .

We perform the object detection experiments with a YOLOv5s instance that uses the CSPDarknet53 [48] backbone. The model is trained for 100 epochs, using SGD with momentum. We set the learning rate to 0.01 in all experiments and use a weight decay of $5 \cdot 10^{-4}$, with a warm-up stage of 3 epochs, where the learning rate is increasing from $3 \cdot 10^{-6}$ to 10^{-2} . For the LeRaC experiments, we replace the warm-up stage with the actual curriculum method. The momen-

Table 2: Average accuracy rates (in %) over five runs on CIFAR-10, CIFAR-100, ImageNet and Food-101 with various neural models based on different training regimes: conventional, CBS [42], LSCL [11], LeRaC [9], EfficientTrain [50], and CBM (ours). The accuracy of the best training regime in each experiment is highlighted in bold.

Model	Training regime	CIFAR-10	CIFAR-100	ImageNet	Food-101
ResNet-18	conventional	89.20 ± 0.43	65.28 ± 0.16	57.41 ± 0.05	68.31 ± 0.09
	CBS [42]	89.53 ± 0.22	66.41 ± 0.21	55.49 ± 0.20	65.09 ± 0.47
	LSCL [11]	88.28 ± 0.14	67.59 ± 0.25	57.27 ± 0.36	69.47 ± 0.30
	LeRaC [9]	89.56 ± 0.16	66.02 ± 0.17	57.86 ± 0.20	69.57 ± 0.07
	EfficientTrain [50]	89.51 ± 0.13	68.13 ± 0.12	57.77 ± 0.17	67.96 ± 0.66
	CBM (ours)	90.48 ± 0.12	67.90 ± 0.08	59.50 ± 0.30	70.77 ± 0.23
Wide-ResNet-50	conventional	91.22 ± 0.24	68.14 ± 0.16	60.97 ± 0.30	67.54 ± 0.66
	CBS [42]	89.05 ± 1.00	65.73 ± 0.36	53.30 ± 1.53	58.95 ± 1.80
	LSCL [11]	91.24 ± 0.13	68.34 ± 0.49	62.70 ± 0.23	69.20 ± 0.48
	LeRaC [9]	91.58 ± 0.16	69.38 ± 0.26	61.98 ± 0.42	67.96 ± 0.35
	EfficientTrain [50]	91.03 ± 0.28	69.14 ± 0.20	62.44 ± 0.11	70.33 ± 0.54
	CBM (ours)	92.67 ± 0.09	70.90 ± 0.31	63.30 ± 0.54	72.09 ± 0.14
CvT-13	conventional	93.56 ± 0.05	77.80 ± 0.16	70.71 ± 0.35	85.22 ± 0.11
	CBS [42]	85.85 ± 0.15	62.35 ± 0.48	68.41 ± 0.13	81.41 ± 0.42
	LSCL [11]	93.91 ± 0.20	78.63 ± 0.12	70.96 ± 0.30	82.49 ± 0.10
	LeRaC [9]	94.15 ± 0.03	78.93 ± 0.05	71.34 ± 0.08	86.05 ± 0.08
	EfficientTrain [50]	94.50 ± 0.17	78.20 ± 0.34	71.64 ± 0.33	85.53 ± 0.14
	CBM (ours)	95.00 ± 0.10	81.14 ± 0.21	71.82 ± 0.20	87.96 ± 0.08

tum is set to 0.8 during warm-up, and 0.94 afterwards. The masking schedule is the same as the one employed for the recognition models, but the maximum masking ratio is 0.3 and the number of patches is 16×16 .

Environment. The experiments are conducted on a server with two Intel Xeon v4 3.0GHz CPUs, 256GB of RAM, and four Nvidia GeForce GTX 1080 GPUs, each with 11GB of VRAM.

Evaluation. The object recognition models are evaluated in terms of the classification accuracy. We repeat each experiment five times and report the average accuracy and the standard deviation.

For the object detection models, we compute the Average Precision (AP) on each class and report the performance over all classes, *i.e.* the mean Average Precision (mAP). Following standard evaluation protocols, we consider that an object is correctly detected if the intersection over union (IoU) between the predicted and ground-truth bounding boxes is at least 0.5. Hence, the reported measure is mAP@IoU=0.5.

4.2 Results

Object recognition. In Table 2, we report the object recognition results with six training regimes (conventional, CBS, LSCL, LeRaC, EfficientTrain and CBM) on CIFAR-10, CIFAR-100, ImageNet and Food-101. We report the average accuracy rates and the standard deviations over five runs with each model and training regime.

On CIFAR-10, the only training regime that makes ResNet-18 surpass the 90% threshold is CBM, boosting the performance by 1.28% over the baseline. We observe similar gains brought by our method (CBM) to the Wide-ResNet-50 (+1.45%) and CvT-13 (+1.44%) architectures. In contrast, CBS is only able to bring improvements for ResNet-18, and LSCL is only able to improve CvT-13. EfficientTrain seems to be slightly better than CBS and LSCL, although it degrades the performance of Wide-ResNet-50. LeRaC is the top competitor, being consistent across all architectures, but its accuracy gains are always below 1% on CIFAR-10.

On CIFAR-100, we observe that the CBS regime is not helpful for Wide-ResNet-50 and CvT-13, degrading their performance rates by significant margins. LeRaC proves to be a worthy competitor, always boosting the accuracy rate of the baseline. Its performance boosts are

Table 3: Mean Average Precision (mAP) at an IoU threshold of 0.5 on PASCAL VOC 2007+2012 for the YOLOv5 [26] model based on different training regimes: conventional, CBS [42], LSCL [11], LeRaC [9], EfficientTrain [50], and CBM (ours). The best result is highlighted in bold.

Training regime	mAP@0.5
conventional	0.832 ± 0.006
CBS [42]	0.829 ± 0.003
LSCL [11]	0.833 ± 0.005
LeRaC [9]	0.846 ± 0.004
EfficientTrain [50]	0.839 ± 0.003
CBM (ours)	0.847 ± 0.001

above 1% for Wide-ResNet-50 and CvT-13. EfficientTrain is the best regime for ResNet-18, but it only ranks third and fourth for Wide-ResNet-50 and CvT-13. In contrast, CBM brings the highest accuracy gains for Wide-ResNet-50 and CvT-13, remarkably surpassing all its competitors by more than 1.52% for Wide-ResNet-50 and 2.21% for CvT-13, respectively.

ImageNet is the most challenging benchmark included in our experiments. Still, the trends observed on CIFAR-10 and CIFAR-100 also apply to ImageNet. For example, CBS degrades the performance, just as before. In contrast, LSCL, LeRaC and EfficientTrain outperform the conventional training regime, regardless of the underlying architecture. Yet, CBM stands out as the best training regime, outperforming all competitors by more than 1.64% for ResNet-18.

The results reported on Food-101 are mostly consistent with the results reported on ImageNet. CBS is often harmful, while LSCL, LeRaC and EfficientTrain are usually able to increase performance. Still, CBM exhibits the highest gains for all three architectures, showing significant differences with respect to its competitors. For instance, CBM surpasses the second-best regime for CvT-13 by 1.91%.

In summary, the object recognition experiments point towards some generic conclusions. The CBS [42] training regime is only useful in some cases. LSCL [11] and EfficientTrain [50] are usually helpful, although they sometimes reduce performance. LeRaC [9] brings consistent accuracy gains, but the improvements are often below 1%. In contrast, the proposed regime, CBM, outperforms the other training regimes on all four data sets and for all three architec-

Table 4: Average accuracy rates (in %) over five runs on CIFAR-10, CIFAR-100 and ImageNet with various curriculum schedules. The accuracy of the best regime for each model and data set pair is highlighted in bold.

Data set	Curriculum schedule	Model		
		ResNet-18	WRResNet-50	CvT-13
CIFAR-10	-	89.20±0.43	91.22±0.24	93.56±0.05
	Constant	89.41±0.08	92.08±0.06	94.36±0.09
	Linear	90.24±0.05	92.39±0.11	94.71±0.05
	Log	90.31±0.09	92.61±0.08	94.49±0.07
	Exp	90.35±0.06	91.80±0.15	94.43±0.12
	Lin repeat	90.48±0.12	92.67±0.09	95.00±0.10
CIFAR-100	-	65.28±0.16	68.14±0.16	77.80±0.16
	Constant	66.58±0.22	68.64±0.16	77.37±0.32
	Linear	66.46±0.28	70.51±0.20	78.94±0.25
	Log	67.51±0.12	70.75±0.30	79.22±0.13
	Exp	67.27±0.07	68.82±0.23	79.11±0.11
	Lin repeat	67.90±0.08	70.90±0.31	81.14±0.21
ImageNet	-	57.41±0.05	60.97±0.30	70.71±0.35
	Constant	57.82±0.13	61.25±0.22	70.14±0.07
	Linear	58.93±0.18	63.61±0.97	70.92±0.08
	Log	59.17±0.14	62.82±0.29	70.78±0.13
	Exp	58.02±0.23	62.10±0.05	71.35±0.14
	Lin repeat	59.50±0.30	63.30±0.54	71.82±0.20

tures. We performed Cochran’s Q statistical testing for conventional training versus CBM. The statistical tests indicate that our performance gains (on all data sets and architectures) are significant, at a p-value of 0.001.

Object detection. We explore the applicability of the curriculum learning methods on the object detection task, reporting the mAP scores of YOLOv5s on PASCAL VOC [13], while alternating between the six training regimes. The corresponding results are reported in Table 3. The object detection results indicate that LeRaC, EfficientTrain and CBM lead to sizeable improvements, while CBS seems to degrade performance. Our method obtains an mAP of 0.847, outperforming all other regimes. Hence, the object detection results confirm that our approach is a viable curriculum learning strategy.

4.3 Ablation Results

Curriculum schedules. We present additional results with various curriculum schedules in Table 4. We include the vanilla training regime as a baseline. The constant schedule makes our framework equivalent to that of He *et al.* [21]. Although He *et al.* [21] showed that masked auto-encoders reach optimal results in self-supervised learning, we observe that their training regime leads to marginal improvements in supervised learning. However, there are multiple curriculum learning schedules that bring significant performance improvements over the vanilla training regime and the constant schedule, for each combination of model and data set. Thus, the lower accuracy rates of the constant schedule can be attributed to the lack of easy-to-hard curriculum in this type of schedule. Among the considered schedules, the *linear*, *log* and *linear repeat* schedules exhibit the highest performance gains, but the best choice is clearly *linear repeat*. This observation motivates our use of the linear repeat schedule in the experiments presented in Tables 2 and 3.

Ablation of gradient masking and curriculum. We first study the effect of our gradient-based masking procedure, replacing it with masking based on the uniform distribution of the patches. We jointly assess the influence of the curriculum schedule, as opposed to using a constant masking ratio during the whole training process. We present these experiments on CIFAR-100 in Table 5. Without the curriculum, the gradient-based masking brings accuracy gains for ResNet-18 and

Table 5: Ablation study of the gradient-based masking and the curriculum schedule on CIFAR-100. We alternatively deactivate the gradient-based masking and the curriculum to determine their individual benefits.

Model	Gradient masking	Curriculum	Accuracy
ResNet-18	✗	✗	65.28 ± 0.16
	✓	✗	66.58 ± 0.22
	✗	✓	67.75 ± 0.17
	✓	✓	67.90 ± 0.08
Wide-ResNet-50	✗	✗	68.14 ± 0.16
	✓	✗	68.64 ± 0.16
	✗	✓	70.45 ± 0.11
	✓	✓	70.90 ± 0.31
CvT-13	✗	✗	77.80 ± 0.16
	✓	✗	78.37 ± 0.32
	✗	✓	80.02 ± 0.16
	✓	✓	81.14 ± 0.21

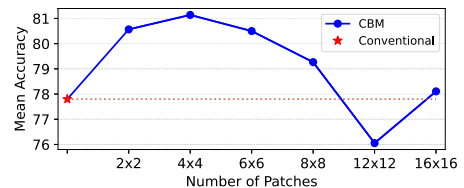


Figure 3: Varying the number of masking patches of the linear repeat schedule, for CvT-13 on CIFAR-100. There are multiple configurations that surpass the baseline.

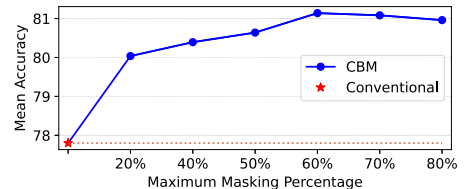


Figure 4: Varying the maximum masking ratio of the linear repeat schedule, for CvT-13 on CIFAR-100. All hyperparameter choices outperform the baseline.

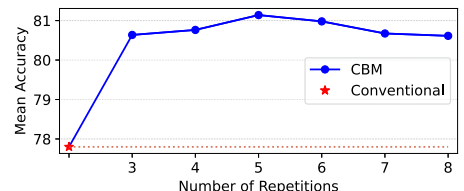


Figure 5: Varying the number of repetitions of the linear repeat schedule, for CvT-13 on CIFAR-100. All hyperparameter values lead to better results than the baseline.

Wide-ResNet-50, but not for CvT-13. We performed Cochran’s Q statistical tests to assess the significance of gradient-based masking. For ResNet-18 and Wide-ResNet-50, the gains are significant, at p-values of 0.001 and 0.01, respectively. Replacing the constant schedule with the proposed curriculum schedule (linear repeat) improves the accuracy rates of all models. When the easy-to-hard curriculum is activated, the gradient-based masking increases the accuracy, but the improvements seem low. While the results indicate that both our contributions are effective, our curriculum approach has a much higher positive impact on performance. We report the highest gains when combining the gradient-based masking with our curriculum learning schedule, confirming the effectiveness of our joint design. From a practical point of view, we thus judge our contributions as relevant.

Table 6: Results for CvT-13 on CIFAR-100 with CutMix, Masking and CutMix+Masking.

Augmentation	Baseline	Constant	Linear	Lin Repeat
No augmentation	77.80 \pm 0.16	-	-	-
CutMix	-	74.94 \pm 0.52	79.58 \pm 0.17	80.87 \pm 0.63
Masking	-	77.37 \pm 0.32	78.94 \pm 0.25	81.14 \pm 0.21
CutMix+Masking	-	79.56 \pm 0.12	79.78 \pm 0.26	81.34 \pm 0.25

Table 7: Accuracy rates (in %) on CIFAR-10 with the ViT model based on two training regimes: conventional and CBM.

Data set	ViT (conventional)	ViT (CBM)
CIFAR-10	94.55 \pm 0.16	95.32 \pm 0.21

Ablation of the linear repeat schedule. We conduct additional ablation studies for our best schedule, namely linear repeat. In Figure 3, we present results with CvT-13 [52] on CIFAR-100 [29] while varying the number of patches for the linear repeat schedule. We observe that any number of patches between 2×2 and 8×8 leads to better results than the baseline. This confirms that our approach is fairly robust to suboptimal configurations of the number of patches. In Figure 4, we show results with CvT-13 [52] on CIFAR-100 [29] while varying the maximum masking ratio for the linear repeat schedule. Remarkably, all configurations produce better results than the baseline. However, we generally observe larger improvements for higher maximum masking ratios. In Figure 5, we present additional results on CIFAR-100 [29] with linear repeat, using different repeat intervals. Once again, we notice that all hyperparameter choices outperform the baseline. In summary, the ablation results show that CBM based on linear repeat is very robust to suboptimal hyperparameter choices.

4.4 Additional Results

Augmentation versus curriculum. We underline that the constant schedule is equivalent to simple data augmentation via masking patches. As shown in Table 4 from the main article, the differences between the baseline and the constant schedule (which correspond to data augmentation) and the differences between the constant and linear repeat schedules show that curriculum learning brings higher improvements on CIFAR-100 and ImageNet. Moreover, simple data augmentation degrades performance for CvT-13 on CIFAR-100 and ImageNet. This confirms that the reported improvements are mostly due to curriculum learning.

To further show that our curriculum strategy can be applied to other data augmentations, we conduct experiments with CutMix [54] and curriculum learning for CvT-13 on CIFAR-100. As shown in Table 6, the CutMix augmentation does not work at all without the curriculum learning strategies, namely linear and linear repeat. Notably, combining CutMix and masking with the linear repeat curriculum leads to additional gains. These results support our previous observation, namely that the performance gains are rather the effect of employing curriculum learning.

Compatibility with vision transformers. In Table 2, we report results with CvT for a direct comparison with LeRaC [9], one of our main competitors, which also uses CvT. However, our method is not tied to a certain architecture. To demonstrate this, we conduct additional experiments with ViT [12] on CIFAR-10. The results shown in Table 7 confirm that CBM is also useful for ViT.

Qualitative results. In Figure 6, we include a few examples to illustrate the benefits brought by our curriculum learning method on the classification task. We can see that the model trained with our strategy is able to better discriminate between similar classes. Indeed, the

Method	Label	Image
Ground-Truth	beer bottle	
Conventional	soda bottle	
CBS	soda bottle	
LeRaC	soda bottle	
CBM (ours)	beer bottle	
Ground-Truth	king penguin	
Conventional	king penguin	
CBS	snorkel	
LeRaC	snorkel	
CBM (ours)	king penguin	
Ground-Truth	tabby cat	
Conventional	boa constrictor	
CBS	tailed frog	
LeRaC	tabby cat	
CBM (ours)	tabby cat	
Ground-Truth	cougar	
Conventional	lion	
CBS	cougar	
LeRaC	cougar	
CBM (ours)	cougar	

Figure 6: Qualitative results for the image recognition task, comparing the vanilla regime with CBS [42], LeRaC [9], and CBM (ours). The samples are taken from ImageNet [40]. In the illustrated examples, we can observe that our curriculum learning method helps the model to better distinguish between similar classes. Best viewed in color.

first three image samples show that our method outperforms the other ones in the shown cases. As for the last example in Figure 6, CBM aligns with the other curriculum learning strategies and predicts the correct class for the corresponding image.

5 Conclusion

In this paper, we proposed a new curriculum learning method based on masking input patches. Our method uses a gradient-based masking procedure in conjunction with a gradually increasing masking ratio to create an easy-to-hard curriculum without having to estimate the difficulty of training samples. To demonstrate the effectiveness of CBM, we carried out a comprehensive set of experiments, considering multiple neural architectures and data sets. Our empirical results show that CBM surpasses the vanilla training regime, as well as four state-of-the-art curriculum learning strategies [9, 11, 42, 50]. Moreover, we performed ablation experiments to justify our design integrating the gradient-based masking and the easy-to-hard curriculum.

References

- [1] D. Amodei, S. Anantharayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, et al. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *Proceedings of ICML*, pages 173–182, 2016.
- [2] W. G. C. Bandara, N. Patel, A. Gholami, M. Nikkiah, M. Agrawal, and V. M. Patel. AdaMAE: Adaptive Masking for Efficient Spatiotemporal Learning With Masked Autoencoders. In *Proceedings of CVPR*, pages 14507–14517, 2023.
- [3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum Learning. In *Proceedings of ICML*, pages 41–48, 2009.
- [4] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *Proceedings of ECCV*, pages 446–461, 2014.
- [5] M. Burduja and R. T. Ionescu. Unsupervised Medical Image Alignment with Curriculum Learning. In *Proceedings of ICIP*, pages 3787–3791, 2021.

- [6] A. Caubrière, N. A. Tomashenko, A. Laurent, E. Morin, N. Camelin, and Y. Estève. Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability. In *Proceedings of INTERSPEECH*, pages 1198–1202, 2019.
- [7] X. Chen and A. Gupta. Webly Supervised Learning of Convolutional Networks. In *Proceedings of ICCV*, pages 1431–1439, 2015.
- [8] V. Cirik, E. Hovy, and L.-P. Morency. Visualizing and Understanding Curriculum Learning for Long Short-Term Memory Networks. *arXiv preprint arXiv:1611.06204*, 2016.
- [9] F.-A. Croitoru, N.-C. Ristea, R. T. Ionescu, and N. Sebe. Learning Rate Curriculum. *International Journal of Computer Vision*, 2024.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, pages 4171–4186, 2019.
- [11] Ü. Dogan, A. A. Deshmukh, M. B. Machura, and C. Igel. Label-similarity curriculum learning. In *Proceedings of ECCV*, pages 174–190, 2020.
- [12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of ICLR*, 2021.
- [13] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [14] Y. Fan, R. He, J. Liang, and B.-G. Hu. Self-Paced Learning: An Implicit Regularization Perspective. In *Proceedings of AAAI*, pages 1877–1883, 2017.
- [15] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked autoencoders as spatiotemporal learners. In *Proceedings of NeurIPS*, pages 35946–35958, 2022.
- [16] M.-I. Georgescu, E. Fonseca, R. T. Ionescu, M. Lucic, C. Schmid, and A. Arnab. Audiovisual masked autoencoders. In *Proceedings of ICCV*, 2023.
- [17] M. Gong, H. Li, D. Meng, Q. Miao, and J. Liu. Decomposition-based evolutionary multiobjective optimization to self-paced learning. *IEEE Transactions on Evolutionary Computation*, 23(2):288–302, 2019.
- [18] L. Gui, T. Baltrušaitis, and L.-P. Morency. Curriculum Learning for Facial Expression Recognition. In *Proceedings of FG*, pages 505–511, 2017.
- [19] G. Hacohen and D. Weinshall. On The Power of Curriculum Learning in Training Deep Networks. In *Proceedings of ICML*, pages 2535–2544, 2019.
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of CVPR*, pages 770–778, 2016.
- [21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of CVPR*, pages 16000–16009, 2022.
- [22] R. T. Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari. How Hard Can It Be? Estimating the Difficulty of Visual Search in an Image. In *Proceedings of CVPR*, pages 2157–2166, 2016.
- [23] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-Paced Curriculum Learning. In *Proceedings of AAAI*, pages 2694–2700, 2015.
- [24] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *Proceedings of ICML*, pages 2304–2313, 2018.
- [25] A. Jiménez-Sánchez, D. Mateus, S. Kirchhoff, C. Kirchhoff, P. Biberthaler, N. Navab, et al. Medical-based Deep Curriculum Learning for Improved Fracture Classification. In *Proceedings of MICCAI*, pages 694–702, 2019.
- [26] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, and et. al. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. 2022. URL <https://doi.org/10.5281/zenodo.7347926>.
- [27] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *Proceedings of ICLR*, 2018.
- [28] T. Kocmi and O. Bojar. Curriculum Learning and Minibatch Bucketing in Neural Machine Translation. In *Proceedings of RANLP*, pages 379–386, 2017.
- [29] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [30] M. Kumar, B. Packer, and D. Koller. Self-Paced Learning for Latent Variable Models. In *Proceedings of NIPS*, volume 23, pages 1189–1197, 2010.
- [31] H. Li, M. Gong, D. Meng, and Q. Miao. Multi-Objective Self-Paced Learning. In *Proceedings of AAAI*, pages 1802–1808, 2016.
- [32] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He. Scaling language-image pre-training via masking. In *Proceedings of CVPR*, pages 23390–23400, 2023.
- [33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al. Microsoft COCO: Common Objects in Context. In *Proceedings of ECCV*, pages 740–755, 2014.
- [34] C. Liu, S. He, K. Liu, and J. Zhao. Curriculum Learning for Natural Answer Generation. In *Proceedings of IJCAI*, pages 4223–4229, 2018.
- [35] N. Madan, N.-C. Ristea, K. Nasrollahi, T. B. Moeslund, and R. T. Ionescu. Cl-mae: Curriculum-learned masked autoencoders. In *Proceedings of WACV*, pages 2492–2502, 2024.
- [36] A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In *Proceedings of CVPR*, pages 5492–5500, June 2015.
- [37] E. A. Platanios, O. Stretcu, G. Neubig, B. Poczós, and T. Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of NAACL*, pages 1162–1172, 2019.
- [38] S. Ranjan and J. H. L. Hansen. Curriculum Learning Based Approaches for Noise Robust Speaker Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26:197–210, 2018.
- [39] N.-C. Ristea and R. T. Ionescu. Self-paced ensemble learning for speech and audio classification. In *Proceedings of INTERSPEECH*, pages 2836–2840, 2021.
- [40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [41] M. Shi and V. Ferrari. Weakly Supervised Object Localization Using Size Estimates. In *Proceedings of ECCV*, pages 105–121, 2016.
- [42] S. Sinha, A. Garg, and H. Larochelle. Curriculum by smoothing. In *Proceedings of NeurIPS*, pages 21653–21664, 2020.
- [43] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe. Curriculum self-paced learning for cross-domain object detection. *Computer Vision and Image Understanding*, 204:103–166, 2021.
- [44] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.
- [45] V. I. Spitzkovsky, H. Alshawi, and D. Jurafsky. Baby steps: How “less is more” in unsupervised dependency parsing. In *Proceedings of NIPS*, 2009.
- [46] Y. Tay, S. Wang, A. T. Luu, J. Fu, M. C. Phan, X. Yuan, et al. Simple and Effective Curriculum Pointer-Generator Networks for Reading Comprehension over Long Narratives. In *Proceedings of ACL*, pages 4922–4931, 2019.
- [47] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Proceedings of NeurIPS*, 2022.
- [48] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and L.-H. Yeh. CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of CVPRW*, pages 390–391, 2020.
- [49] X. Wang, Y. Chen, and W. Zhu. A Survey on Curriculum Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022.
- [50] Y. Wang, Y. Yue, R. Lu, T. Liu, Z. Zhong, S. Song, et al. EfficientTrain: Exploring Generalized Curriculum Learning for Training Visual Backbones. In *Proceedings of ICCV*, pages 5852–5864, 2023.
- [51] J. Wei, A. Suriawinata, B. Ren, X. Liu, M. Lisovsky, L. Vaickus, et al. Learn like a Pathologist: Curriculum Learning by Annotator Agreement for Histopathology Image Classification. In *Proceedings of WACV*, pages 2472–2482, 2021.
- [52] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, et al. CvT: Introducing Convolutions to Vision Transformers. In *Proceedings of ICCV*, pages 22–31, 2021.
- [53] L. Wu, F. Tian, Y. Xia, Y. Fan, T. Qin, L. Jian-Huang, et al. Learning to Teach with Dynamic Loss Functions. In *Proceedings of NeurIPS*, volume 31, pages 6467–6478, 2018.
- [54] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *Proceedings of ICCV*, pages 6023–6032, 2019.
- [55] S. Zagoruyko and N. Komodakis. Wide Residual Networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [56] M. Zhang, Z. Yu, H. Wang, H. Qin, W. Zhao, and Y. Liu. Automatic Digital Modulation Classification Based on Curriculum Learning. *Applied Sciences*, 9(10), 2019.
- [57] W. Zhang, W. Wei, W. Wang, L. Jin, and Z. Cao. Reducing BERT Computation by Padding Removal and Curriculum Learning. In *Proceedings of ISPA*, pages 90–92, 2021.
- [58] S. Zhou, J. Wang, D. Meng, X. Xin, Y. Li, Y. Gong, et al. Deep self-paced learning for person re-identification. *Pattern Recognition*, 76: 739–751, 2018.