RoBIC: A benchmark suite for assessing classifiers robustness

Thibault Maho, Benoît Bonnet, Teddy Furon, Erwan Le Merrer

RoBIC: A BENCHMARK SUITE FOR ASSESSING CLASSIFIERS ROBUSTNESS

Thibault Maho, Benoît Bonnet, Teddy Furon[†] *and Erwan Le Merrer*

Univ. Rennes, Inria, CNRS, IRISA, Rennes France

ABSTRACT

Many defenses have emerged with the development of adversarial attacks. Models must be objectively evaluated accordingly. This paper systematically tackles this concern by proposing a new parameter-free benchmark we coin RoBIC. RoBIC fairly evaluates the robustness of image classifiers using a new *half-distortion* measure. It gauges the robustness of the network against white and black box attacks, independently of its accuracy. RoBIC is faster than the other available benchmarks. We present the significant differences in the robustness of 16 recent models as assessed by RoBIC.

We make this benchmark publicly available for use and contribution at https://gitlab.inria.fr/tmaho/robustness_benchmark.

***Index Terms*—** Benchmark, adversarial examples, model robustness, half-distortion measure.

1. INTRODUCTION

Deep learning models are vulnerable to adversarial perturbations. This is especially true in image classification in computer vision. This weakness is unfortunatley undermining the developement of 'Artificial Intelligence'. In particular, adversarial attacks are a serious threat for security oriented applications. Attackers willing to bypass security countermeasures might use the deep learning models as the weakest link. A deluge of research papers now propose defenses to block such an attacker, and adaptive attacks against these defenses. This is an endless arms race, and systematic benchmarks to evaluate the state of the threat are greatly required.

It is currently extremely difficult to have a clear view on what is truly working in this domain. The clich is that no two papers report the same statistics for the same attack against the same model over the same image set. This is mostly due to that an attack is an algorithm with many parameters. Its power is indeed highly dependent of these parameters. These values are rarely specified in research papers.

There exist benchmarks in the litterature, such as ARES [1], RobustBench [2], RobustVision [3], ADBD [4]. They aim at providing a better understanding of the robustness of image classifiers. Yet, they fall short because their slowness prevents them from tackling large image dataset like ImageNet. They only operate on CIFAR-10 or MNIST. Also, they resort to attacks which are not all state-of-the-art.

This paper proposes RoBIC, to consider these concerns and develop a benchmark tool to measure the robustness of image classifiers in a modern setup.

2. DIFFICULTIES

This sections explains the difficulties for setting up a benchmark measuring the robustness of image classifiers.

2.1. Notation

An attack is a process forging an image $I_a = \mathcal{A}(I_o, M, \Pi)$, where I_o is the original image, M is the target model, and Π is a set of attack parameters. The ground truth label of I_o is denoted by y_o. The boolean function $\mathbb{1}(I_a, y_o) = [M(I_a) \neq y_o]$ tells whether the attack deludes classifier M in the untargeted attack scenario: the prediction $M(I_a)$ is not the ground truth. The distortion between I_o and I_a is denoted by $d(I_a, I_o)$.

Some statistics like the probability of success and the average distortion are extracted from the adversarial images forged from the test set. They depend on the attack $\mathcal{A}$ and its set of parameters Π. Therefore, it can not play the role of a measure of robustness of a given model. The first difficulty is to get rid off the impact of parameters Π.

2.2. The best effort mode

The parameters Π have a huge impact on the power of an attack. For instance, some attacks like FGSM [5], I-FGSM [6], PGD [6] are distortion constrained in the sense that Π is strongly connected to a distortion budget. If this budget is small, the probability of the success of the attack is small. If it is large, this probability is close to 1 but the distortion is too big. Hence, it is hard to find the best setting to make these attacks competitive. Our strategy, so-called 'best effort mode', reveals the intrinsic power of an attack by finding the best setting for any image: $I_a = \mathcal{A}(I_o, M, \Pi^\star)$ with

$$\Pi^\star = \arg \min_{\Pi : \mathbb{1}(\mathcal{A}(I_o, M, \Pi), y_o) = 1} d(\mathcal{A}(I_o, M, \Pi), I_o). \quad (1)$$

The best effort mode makes the measurement of the robustness independent from an arbitrary global setting Π. Yet, it is

[†] Thanks to ANR and AID for funding Chaire IA *SAIDA* (ANR_20-CHIA-0011-01).

costly in terms of computations. Attacks with few parameters are preferred since the search space is smaller.

2.3. Worst case attacks

A second difficulty is to make the robustness score independent of the attack. Ideally, we would like to know the worst case attack to certify the robustness of a model. An option proposed by benchmarks `RobustVision` [3] and `ARES` [1] is to consider a set of $J = 11$ attacks as outlined in table 1. This is again costly as each image of the test set has to be attacked J times. Yet, a benchmark happens to be useful if it is fast enough so to assess the robustness of many models. The best effort mode over an ensemble of attacks is out of reach. This is the reason why we need to focus on fast worst case attacks in the sense that they achieve their best effort mode within limited complexity. Section 4 focuses on these attacks.

2.4. The choice of the metric

The game between attack $\mathcal{A}$ and model M over the test set is summarized by the operating characteristic $D \to P(D)$ relating the distortion D and the probability of success $P(D)$:

$$P(D) := n^{-1} \sum_{i:d(I_{a,i},I_{o,i})\leq D}^{n} \mathbb{1}(I_{a,i}, y_{o,i}). \tag{2}$$

In other words, $P(D)$ is the fraction of images that the attack succeeded to hack within a distortion budget D. Many benchmarks gauge the robustness by $P(D_b)$ at an arbitrary distortion D_b: *e.g.* `RobustBench` [2] score is $P(D = 0.5)$. This measure is pointwise and dependent on $\eta(0)$.

3. THE BENCHMARK

This section justifies the recommendations made in our benchmark and defines the measure of robustness.

Pixel domain. Our benchmark is dedicated to image classification. As a consequence, the distortion is defined on the pixel domain: An image I is defined in the space $[\![0, 255]\!]^n$ with $n = 3RC$ pixels for 3 color channels, R rows and L columns. Most papers in the field measure distortion after the transformation of the image in a tensor $x \in \mathcal{X}^n$. This is a mistake preventing a fair comparison: for most models $\mathcal{X} = [0, 1]$, but for some others $\mathcal{X} = [-1, 1]$ or $\mathcal{X} = [-3, 3]$.

We outline that an adversarial image is above all an image, *i.e.* a discrete object $I_a \in [\![0, 255]\!]^n$. Again, most attacks output a continuous tensor $x_a \in \mathcal{X}^n$, neglecting the quantization. This is a mistake: in real-life, the attacker has no access to x_a, which is an auxiliary data internal of the model.

Distortion. The distortion is defined as the root mean square error: $d(I_a, I_o) := \|I_a - I_o\|_2/\sqrt{n}$. This is easily interpretable: if $I_{a,i} = I_{o,i} \pm \epsilon, \forall i \in [\![1, n]\!]$, then $d(I_o, I_a) = \epsilon$. It is easily translated into a PNSR as image processing professionals do: $\mathrm{PSNR} = 48.13 - 20 \log_{10}(d(I_o, I_a))$ dB. Adversarial perturbations usually spread all over the image and have small amplitude like in invisible watermarking. This is a case where measures based on ℓ_2 norm remain good indicators of the quality. A perceptual similarity is obviously better, but more complex and less interpretable.

Test set. The input of the model is a natural and large image. Assessing the robustness of models on specific dataset like MNIST (almost black and white), or on tiny images like CIFAR does not reflect the complexity of the problem. Our benchmark considers natural images of at least 224×224 pixels as provided in ImageNet.

Measure of robustness. Let us define the accuracy function $\eta(D) := 1 - P(D)$. The value $\eta(0)$ is the classical accuracy of the model over original images. Function $\eta(D)$ is by construction non increasing and should converge to 0 as the distortion D increases. After observing many accuracy functions η for different models and attacks, we notice that they share the same prototype:

$$\eta(D) = \eta(0)\, e^{-\lambda D} \quad \text{with } \lambda \in \mathbb{R}^+. \tag{3}$$

Like in nuclear physics, we define the half-distortion $D_{1/2}$ as the distortion needed to reduce to half the initial accuracy:

$$\eta(D_{1/2}) = \eta(0)/2, \quad D_{1/2} = \lambda^{-1} \log(2). \tag{4}$$

This approximation is verified experimentally with an average coefficient of determination R^2 of 99%. The half-distortion $D_{1/2}$ will be the keystone of the proposed metric of robustness. A model is then characterized by three separated concepts: its generalization ability $\eta(0)$ and its robustnesses $D_{1/2}$ against black-box and white-box attacks.

4. FAST ATTACKS

The recent trend in adversarial examples is to design fast attacks with state-of-the-art performances.

4.1. Fast black-box attacks

In the black-box decision based setup, the attacker can query a model and observes the predicted class. The complexity of the attack is gauged by the number of queries K needed to find an adversarial image of low distortion.

There has been a huge improvement on the amount of queries recently. Brendel *et al.* report in the order of one million of queries for one image in one of the first decision based black-box `BA` [7, Fig. 6]. Then, the order of magnitude went down to tens of thousands [8, Fig. 4] [9, Fig. 5] and even some thousands in [10, Fig. 2]. Current benchmarks use others black-box attacks, which are either decision-based (`Square Attack` [11] in `RobustBench` [2] is score-based), or not state-of-the-art (like Gaussian noise in `RobustVision` [3], or `BA` [7] in `ARES` [1]).

SurFree [12] and RayS [4] are the only *decision-based* papers with less than one thousand of calls on ImageNet. Yet, RayS [4] is designed to minimize the ℓ_∞ distortion, whereas SurFree [12] targets ℓ_2. Sect. 5 investigates which attack is the best candidate for a fast benchmark.

4.2. Fast white-box attacks

In the white-box setup, the attacker can compute a loss function and its gradient thanks to auto-differentiation and back-propagation. The complexity is usually gauged by the number of gradient computations. Current benchmarks use different white-box attacks: RobustBench [2] relies on PGD [6] (with 2 parameters Π), RobustVision [3] use DeepFool [13], and ARES [1] CW [14].

Again, the need for powerful but fast attacks is of utmost importance for a practical benchmark. A promising attack is BP [15] designed for low complexity budget. Its first stage finds an adversarial example as quickly as possible. It is nothing more than a gradient descent of the loss L with acceleration. At iteration $t+1$:

$$I_a^{(t+1)} = I_a^{(t)} - \alpha\gamma(t+1)\eta\left(\nabla L(I_a^{(t)})\right), \tag{5}$$

where $I_a^{(0)} = I_o$, $\eta(x) = x/\|x\|_2$, and $\gamma(t)$ is a series of increasing values, hence the acceleration. Stage 1 finishes when $I_a^{(t+1)}$ becomes adversarial. Stage 2 aims at lowering the distortion while maintaining the image adversarial (see [15]).

We develop a variant to aggressively downsize the number of gradient computations. Parameter α is heuristically set up to 0.03 in [15]. This value is certainly too big for images close to the class boundary and too small for those further away. One costly option is the best effort mode which finds the best α thanks to a line search (see Sect. 2). We propose the following simple method inspired by DeepFool [13]. When applying (5) to the first order approximation of the loss:

$$L(I_o + p) \approx L(I_o) + p^\top \nabla L(I_o), \tag{6}$$

then $\eta\left(\nabla L(I_a^{(t)})\right) = \eta(\nabla L(I_o))$ and BP cancels the loss for

$$\alpha = \frac{L(I_o)}{\|\nabla L(I_o)\|_2 \sum_{j=1}^{\kappa}\gamma(k)} \tag{7}$$

within κ iterations. We fix $\kappa = \lfloor K/3 \rfloor$ where K is the total iteration budget encompassing stages 1 and 2.

Sect. 5 compares these attacks to identify the worst case.

4.3. Quantization

The adversarial samples are quantified in the pixel domain to create images. The first option considers the quantization as a post-processing not interfering with the attack. The second option performs quantization at the end of any iteration.These options are tested on several black and white box attacks. The quantization will be a post-processing for white-box attacks as recommended in [16], whereas the second option give better results on black-box attacks.

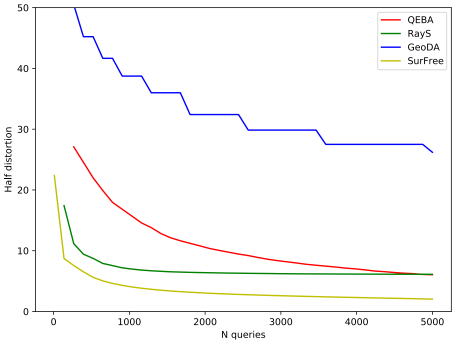

Fig. 1. Evolution of $D_{1/2}$ with the complexity budget for black box setup. Attacks on EfficientNet [17]

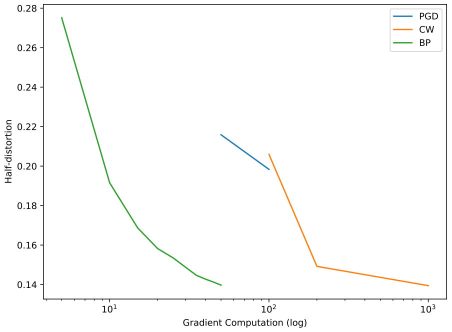

Fig. 2. Evolution of $D_{1/2}$ with the complexity budget for white box setup. Attacks on EfficientNet [17]

5. EXPERIMENTS

All the attacks are run on 1000 ImageNet images from the ILSVRC2012's validation set with size $n = 3 \times 224 \times 224$.

5.1. Selecting the worst case attacks

Black box attacks: Figure 1 compares the evolution of the half-distortion (4) in function of the query amount for four decision-based black-box attacks: SurFree [12], RayS [4], GeoDA [10], and QEBA [9]. SurFree and RayS reach their best effort within 3000 queries, while QEBA and GeoDA do not since their $D_{1/2}$ still decrease after 5000 queries. Yet, SurFree obtains quantified adversarials with much lower distortion. Therefore, our benchmark only needs this attack. The number of queries is kept at 5000 to be sure to reach the optimal value of $D_{1/2}$.

White box attacks: Figure 2 compares three white-box-attacks in the best effort mode: PGD [6], CW [14], and BP [15] with our trick (7). They all reach the same $D_{1/2}$ when given a large complexity budget. Yet, BP converges faster than the others. Our benchmark uses this version of BP to evaluate the white-box-robustness.

5.2. Comparison with other benchmarks

Table 1 lists several benchmarks. Most of them evaluate the robustness as the success-rate under a prescribed ℓ_2 or ℓ_∞ distortion budget. But, these budgets are set arbitrarily or even not constant within the same benchmark for RobustML. Our half-distortion (4) is parameter-free. It returns an accurate, reliable and fair measurement of robustness.

Some benchmarks need many attacks to get a full vision of the robustness: ARES [1] and RobustVision [3] use 11 attacks. This is too time-consuming. On the contrary, ADBD [4] focuses on a single black-box attack, which is indeed outdated. RobustBench [2] condenses four attacks in one measure elegantly: for a given image, if the first simple attack does not succeed within the distortion budget, then the second more complex one is launched *etc.* The total runtime heavily depends on the distortion budget. Yet, black-box and white-box attacks use different mechanisms. Our benchmark reports a measurement for each separately.

5.3. Benchmarking models

Table 2 compares standard models from *timm* [31] and *torchvision* [32] libraries. Here are some intriguing results.

Robustness in white box *vs.* black box. One does not imply the other. Fig. 3 even shows a negative correlation. However, some models escape this rule. For instance, VGG16 is neither robust in black box nor in white box. EfficientNet AdvProp [28] follows the opposite trend. We believe that black-box robustness reveals the complexity of the borders between classes, and white-box robustness indicates how close natural images are from the borders. This highlights the importance of having two different measurements.

The importance of the training procedure. There is on average a factor 20 between the half-distortions in white and black box. This factor drops to 4 and 10 for the models adversarially trained: ResNet50 [18], EfficientNet AdvProp [28].

Table 2 lists four EfficientNet models sharing the same architecture but different training procedures. Their accuracies are similar but there is up to a factor of 2 between the robustnesses. The same holds on the three variants of Resnet50. The gaps in accuracy and robustness are noticeable with standard models from *timm* [31] and *torchvision* [32]. It is even more visible with adversarial training from [18]: the gain in robustness is impressive but at the cost of a big drop in accuracy.

Benchmark	Domain	Nb. attacks	Measures	Runtime
RoBIC	$[\![0,255]\!]^n$	1 WB + 1 BB	Half-distortion ℓ_2	43s
RobustBench [2]	$[0,1]^n$	3 WB + 1 BB	Success-Rate for fixed budget (ℓ_2 or ℓ_∞)	48s
ADBD [4]	$[0,1]^n$	1 BB	Distance ℓ_∞	360s
RobustVision [3]	$[0,1]^n$	6 WB + 5 BB	Median Distance ℓ_2	200s
ARES [1]	$[0,1]^n$	5 WB + 10 BB	Success-Rate vs Budget (ℓ_2, ℓ_∞ or queries)	Too long

Table 1. Benchmarks Comparison. Average Runtimes per ImageNet Image with ResNet50 [18].

Model	Parameters	Accuracy	$D_{1/2}$	
	(millions)	$\eta(0)$	white box	black box
AlexNet [19]	62.38	56.8	0.19	2.17
CSPResNeXt50 [20]	20.57	84.6	0.13	4.48
DualPathNetworks 68b [21]	12.61	83.8	0.08	3.82
MixNet Large [22]	7.33	84.2	0.12	2.96
MobileNetV2 [23]	5.83	80.1	0.09	2.90
ReXNet 200 [24]	16.37	85.4	0.14	3.89
RegNetY 032 [25]	19.44	85.8	0.11	4.94
SEResNeXt50 32x4d [26]	27.56	**85.9**	0.12	**5.01**
VGG16 [27]	138.00	74.9	0.09	2.44
EfficientNet AdvProp [28]	5.29	84.3	**0.31**	4.35
EfficientNet EdgeTPU Small [17]	5.44	82.8	0.15	3.16
EfficientNet NoisyStudent [29]	5.29	82.7	0.19	2.37
EfficientNet [17]	5.29	82.8	0.17	3.56
ResNet50 (torchvision) [30]	25.56	77.9	0.10	2.77
ResNet50 (timm) [30]	25.56	80.5	0.15	4.35
ResNet50 AdvTrain [18]	25.56	60.8	**2.56**	**9.88**

Table 2. Benchmark of models with 1.000 ImageNet Images

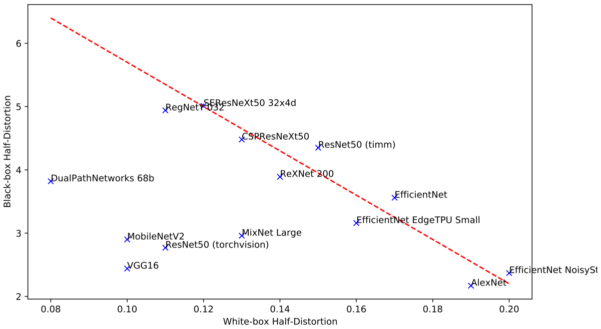

Fig. 3. Black-box $D_{1/2}$ as a function of white-box $D_{1/2}$.

6. CONCLUSION

The paper introduces a rigorous benchmark based on a new and independent measurement of robustness: the half-distortion. RoBIC is faster than the other benchmarks. This allows to tackle larger images which is more realistic.

In addition to the accuracy, RoBIC gives the black box robustness, and white box robustness. We believe that the first indicates how far away the class boundaries lie from the images whereas the last reflects how curved are the boundaries. As the other benchmarks, two limitations hold: The network must be differentiable to run a white box attack, and deterministic to run a black box attack.

7. REFERENCES

[1] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *CVPR*, 2020.

[2] F. Croce, M. Andriushchenko, V. Sehwag, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "Robustbench: a standardized adversarial robustness benchmark," *arXiv preprint arXiv:2010.09670*, 2020.

[3] Bethge Lab, "Robust vision benchmark," `https://robust.vision`.

[4] J. Chen and Q. Gu, "Rays: A ray searching method for hard-label adversarial attack," in *SIGKDD*, 2020.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[6] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR*, 2017.

[7] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *ICLR*, 2018.

[8] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," in *IEEE S&P*, 2020.

[9] H. Li, X. Xu, X. Zhang, S. Yang, and B. Li, "Qeba: Query-efficient boundary-based blackbox attack," in *CVPR*, 2020.

[10] A. Rahmati, S.-M. Moosavi-Dezfooli, P. Frossard, and H. Dai, "Geoda: a geometric framework for black-box adversarial attacks," in *CVPR*, 2020.

[11] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *ECCV*, 2020.

[12] Thibault Maho, Teddy Furon, and Erwan Le Merrer, "Surfree: a fast surrogate-free black-box attack," *arXiv preprint arXiv:2011.12807*, 2020.

[13] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *CVPR*, 2016.

[14] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *S&P*, 2017.

[15] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *IEEE Transactions on IFS*, vol. 16, 2021.

[16] B. Bonnet, T. Furon, and P. Bas, "What if adversarial samples were digital images?," in *IH&MMSec*, 2020.

[17] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for CNN," in *ICML*, 2019.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[19] A. Krizhevsky, "One weird trick for parallelizing CNN," *CoRR*, vol. abs/1404.5997, 2014.

[20] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *CVPR Workshops*, 2020.

[21] Y. Chenand J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *NIPS*, 2017.

[22] Mingxing Tan and Quoc V. Le, "MixConv: Mixed Depthwise Convolutional Kernels," *arXiv e-prints arxiv:1907.09595*.

[23] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018.

[24] D. Han, S. Yun, B. Heo, and Y. Yoo, "ReXNet: Diminishing Representational Bottleneck on CNN," *arXiv e-prints arXiv:2007.00992*, 2020.

[25] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár, "Designing network design spaces," in *CVPR*, 2020.

[26] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[27] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[28] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le, "Adversarial examples improve image recognition," in *CVPR*, 2020.

[29] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[31] Ross Wightman, "Pytorch image models," `https://github.com/rwightman/pytorch-image-models`, 2019.

[32] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *ACM Multimedia*, 2010.