Classifiers Should Do Well Even on Their Worst Classes

Julian Bitterwolf [1] **Alexander Meinke** [1] **Valentyn Boreiko** [1] **Matthias Hein** [1]

Abstract

The performance of a vision classifier on a given test set is usually measured by its accuracy. For reliable machine learning systems, however, it is important to avoid the existence of areas of the input space where they fail severely. To reflect this, we argue that a single number does not provide a complete enough picture even for a fixed test set, as there might be particular classes or subtasks where a model that is generally accurate performs unexpectedly poorly. Without using new data, we motivate and establish a wide selection of interesting worst-case performance metrics which can be evaluated besides accuracy on a given test set. Some of these metrics can be extended when a grouping of the original classes into superclasses is available, indicating if the model is exceptionally bad at handling inputs from one superclass.

1. Introduction and Motivation

The progress of computer vision in the last decade has, to a large extent, been measured by the accuracy of vision classifiers on the ImageNet-2012 validation set (Russakovsky et al., 2015). Generally, more accurate ImageNet models are known to produce more accurate models on shifted datasets (Recht et al., 2019; Taori et al., 2020; Miller et al., 2021) or when the pre-trained weights are transferred to a new task (Kornblith et al., 2019). However, an ImageNet model with higher accuracy can have drastically reduced accuracy on any specific subclass, and as shown in Balestriero et al. (2022), this can negatively affect potential downstream tasks. For this reason, in a general setting, it can be useful to monitor the worst-case accuracy across all classes as the model might generalize poorly to downstream tasks related to the worst-case class. Furthermore, in certain applications fairness considerations might even make it unethical to sacrifice performance in a specific class for a performance gain on average across all classes. To the best of our knowledge, prior fairness measures such as the ones proposed in Xue et al. (2020); Maity et al. (2021); Mukherjee et al. (2020); Speicher et al. (2018); Bellamy et al. (2018) do not capture this particular notion of fairness. We propose a variety of easy-to-evaluate metrics that move beyond the standard measure of accuracy on the test set.

[1]University of Tübingen. Correspondence to: Julian Bitterwolf <julian.bitterwolf@uni-tuebingen.de>.

ICML 2022 Shift Happens Workshop: Crowdsourcing metrics and test datasets beyond ImageNet., Baltimore, Maryland, USA, 2022.

2. Definitions

Let $f : X \to \Delta(C)$ be a classifier that, for an input x, predicts a probability distribution p_c on the simplex $\Delta(C)$ for $c \in C$. We evaluate on a test set D_T of input-label pairs (x, y) which for simplicity we assume for each class $c \in C$ contains at least one sample with $y = c$. Further, for the sake of notational simplicity, we will write $\sum_{x,y}$ for $\sum_{(x,y)\in D_T}$, as we will refer only to the input-label pairs of the test set.

A standard choice of D_T for an ImageNet-2012 trained classifier is the ImageNet-2012 validation set. Standard accuracy is defined as follows:

$$\mathrm{A}(f) = \frac{\sum_{x,y} \mathbb{1}_{\underset{i}{\operatorname{argmax}}\, f(x)_i = y}}{|D_T|} \,. \tag{1}$$

We argue that standard accuracy is not the only metric we are interested in when evaluating a model. As discussed above, it can be detrimental for specific applications if the classifier has severe weaknesses in specific parts of the in-distribution. To get a better overview of the in-distribution weaknesses, we define several accuracy metrics that consider the worst class or classes.

Worst-class accuracy:

$$\mathrm{WCA}(f) = \min_c \frac{\sum_{x,y} \mathbb{1}_{y=c, \underset{i}{\operatorname{argmax}}\, f(x)_i = y}}{\sum_{x,y} \mathbb{1}_{y=c}} \tag{2}$$

is a special case of the "worst-group error" in Liu et al. (2021), with the groups simply being the classes. The worst-class accuracy can also be seen as the **worst-class recall**, i.e. true positives over the sum of true positives and false negatives in that class. This has the interpretation of the class that the classifier is most likely to "miss".

To illustrate why a system that sacrifices worst-class accuracy for standard accuracy could be quite problematic,

consider a facial recognition system. Effectively, such a system can be considered a multi-class classification system with each registered face being a class. A low $\mathrm{WCA}(f)$ would indicate that some person very rarely gets recognized correctly whenever they use the system. This can be seen as quite unfair.

Worst-class precision:

Similarly to the worst-class recall, we can also define the worst-class precision by changing the denominator:

$$\mathrm{WCP}(f) = \min_c \frac{\sum_{x,y} \mathbb{1}_{y=c, \underset{i}{\operatorname{argmax}} f(x)_i = y}}{\sum_{x,y} \mathbb{1}_{\underset{i}{\operatorname{argmax}} f(x)_i = c}} \,. \quad (3)$$

A low $\mathrm{WCP}(f)$ can be interpreted as other classes often being confused for this worst precision class. If we take the example of a facial recognition system again, this could have quite catastrophic consequences. If, for example, security access is in some way tied to proper recognition and the least-precise class corresponds to a high level of clearance, the system could be much less secure than a high standard accuracy would have the user believe.

Worst-superclass accuracy: In practice, models that were pre-trained on ImageNet often get used for downstream tasks. One example of a potential downstream classification task that does not even require fine-tuning would be to deploy an ImageNet model on a distribution that contains only a subset of the classes. For example, one could use a trained model to only separate different dog breeds. We can test how well an ImageNet model would perform on these restricted classification tasks by grouping different ImageNet classes into super-classes and analyzing the accuracy within each superclass. We will define the worst-superclass accuracy as worst-case performance among all superclasses.

$$\mathrm{WSupCA}(f) = \min_{s \in C_{\mathrm{sup}}} \frac{\sum_{x,y} \mathbb{1}_{y \in s, \underset{i \in s}{\operatorname{argmax}} f(x)_i = y}}{\sum_{x,y} \mathbb{1}_{y \in s}} \quad (4)$$

is the same as WCA (Eq. 2), but here instead of classes, we take a subset of WordNet superclasses C_{sup}. Several such choices superclass partitions are possible for ImageNet and different ones have been used (Geirhos et al., 2019; Santurkar et al., 2019; Wen et al., 2022; Engstrom et al., 2019).

Worst-superclass recall: A very closely related measure to the worst-superclass accuracy is what we call the worst-superclass recall:

$$\mathrm{WSupCR}(f) = \min_{s \in C_{\mathrm{sup}}} \frac{\sum_{x,y} \mathbb{1}_{y \in s, \underset{i}{\operatorname{argmax}} f(x)_i = y}}{\sum_{x,y} \mathbb{1}_{y \in s}} \,. \quad (5)$$

The difference to Eq. (4) is that the models predictions are not restricted to lie inside of the given superclass. Thus, for a given classifier f, the $\mathrm{WSupCR}(f)$ is always less than or equal to the $\mathrm{WSupCA}(f)$. These measures can differ significantly, only if mistakes across different superclasses are common.

As a general example, how $\mathrm{WSupCR}(f)$ can show other biases of the classifier, consider again f to be a facial recognition system. Low $\mathrm{WSupCR}(f)$ would indicate that, for example, members of a particular race are consistently not correctly recognized, which is again biased. Since a model with a high $\mathrm{WSupCA}(f)$ can potentially still suffer from this problem, we propose to monitor both metrics, even if, empirically, they often end up quite close to one another.

Worst n-class accuracy: Instead of studying manually chosen groups of classes (like with the superclasses above), we could instead ask what subset of classes leads to the worst accuracy if we restrict our classifier to them. We will call this the worst-n-class accuracy and define it as follows:

$$\mathrm{WnCA}(f) = \min_{C_n \in \binom{C}{n}} \frac{\sum_{x,y} \mathbb{1}_{y \in C_n, \underset{i \in C_n}{\operatorname{argmax}} f(x)_i = y}}{\sum_{x,y} \mathbb{1}_{y \in C_n}} \,, \quad (6)$$

where $\binom{C}{n}$ denotes the set of all combinations of n elements of the set of all classes C.

Note that, $\mathrm{W1CA} = \mathrm{WCA}$ as well as $\mathrm{W|C|CA} = \mathrm{A}$. In practice, computing Eq. (6) for large n is a combinatorial problem and, thus, computationally infeasible. Instead, we propose to compute it only for $n = 2$. The interpretation of this is which two classes are the hardest to separate from one another, which might indicate that this is a downstream task that the model is not suitable for. This is similar to looking at confusing class pairs of confusion matrix as in (Tsipras et al., 2020), but W2CA might be interesting if for a given model there are 2 classes that are consistently misclassified by both not being among the top-2 predicted classes - which means that either the classifier or the images in 2 classes are faulty.

Worst n-class recall: Instead of restricting the classifier's predictions to a subset of classes, we can also simply collect the n classes with the lowest recall:

$$\mathrm{WnCR}(f) = \min_{C_n \in \binom{C}{n}} \frac{\sum_{x,y} \mathbb{1}_{y \in C_n, \underset{i}{\operatorname{argmax}} f(x)_i = y}}{\sum_{x,y} \mathbb{1}_{y \in C_n}} \,. \quad (7)$$

Despite the similarities between Eq. (6) and Eq. (7), note that the WnCR is much easier to compute, as it does not require solving a combinatorial problem in the case of balanced classes (like for ImageNet). Furthermore, for a given classifier f, the $\mathrm{WnCR}(f)$ monotonically increases in n, while providing a lower bound $\mathrm{WnCR}(f) \leq \mathrm{WnCA}(f)$.

Continuing the same example of a face detector, low $\mathrm{WnCR}(f)$ could show, for instance, that f is underperforming on a group of people, that we didn't think to group manually as a superclass, e.g. people with certain hair color.

Worst-class top-k accuracy: As discussed below, the ImageNet validation dataset has egregious weaknesses, containing vast class overlaps and many mislabelled samples. The effect of these weaknesses on standard accuracy can be attenuated (Beyer et al., 2020; Shankar et al., 2020; Vasudevan et al., 2022) by regarding the top-k accuracy (we use the common value $k = 5$) which is defined as:

$$\mathrm{A@k}(f) = \frac{\sum_{x,y} \mathbb{1}_{y \in \underset{i}{\mathrm{top}}(f(x)_i, k)}}{|D_T|} \,. \tag{8}$$

We propose to also study which ground truth class most often evades the top-k detection via the worst-class top-k accuracy:

$$\mathrm{WCA@k}(f) = \min_c \frac{\sum_{x,y} \mathbb{1}_{y=c, y \in \underset{i}{\mathrm{top}}(f(x)_i, k)}}{\sum_{x,y} \mathbb{1}_{y=c}} \,, \tag{9}$$

where the $\underset{i}{\mathrm{top}}(f(x)_i, k)$ operator returns a set of the indices of k maximal elements of $f(x)$. Similarly, we also define the worst-n-class top-k recall:

$$\mathrm{WnCR@k}(f) = \min_{C_n \in \binom{C}{n}} \frac{\sum_{x,y} \mathbb{1}_{y \in C_n, y \in \underset{i}{\mathrm{top}}(f(x)_i, k)}}{\sum_{x,y} \mathbb{1}_{y \in C_n}} \,. \tag{10}$$

3. Experiments

3.1. Implementation

The implementation is available at `https://github.com/valentyn1boreiko/icml-2022`. Since all metrics only take the output logits of a classifier model as input, all evaluations can be performed quickly when provided the outputs of a standard forward pass on the test set. Solely the calculation of W2CA runs in $\mathcal{O}(|C|^2)$ time, amounting to approximately 2 minutes on the ImageNet validation set.

While choosing the worst combination of n classes used for calculating the worst n-class recall WnCR is efficient for a class-balanced test set, this is not feasible for unbalanced classes. Instead, we propose and use a heuristic approximation that chooses the n classes which perform worst individually (without regard to size) and evaluates recall on the samples from these classes. Note that approximating WnCR with this (like any) choice of n classes provides an upper bound on the true value.

As the superclasses for the computation of the WSupCA and WSupCR, we use Restricted ImageNet (Santurkar et al., 2019), which groups the animals in ImageNet into 9 superclasses, i.e. dog, cat, frog, turtle, bird, monkey, fish, crab, and insect.

We evaluate the three models provided in the ShiftHappens API, which are VGG16 (Simonyan & Zisserman, 2014), ResNet-18 and ResNet-50 (He et al., 2016). Additionally, we show results of a Vision Transformer (Dosovitskiy et al., 2021) using the `vit_base_patch16_224` model provided by `timm` (Wightman, 2019).

3.2. Evaluation results

Table 1 shows the evaluation of the proposed metrics with D_T being the full validation set of ImageNet-2012. While we can see that the evaluated model (ViT) with the best standard accuracy (A) also performs best in most other introduced metrics, we observe that even this high-end model shows relative weaknesses when the examiner knows where to look.

Lowered WSupCA attests that cats are hard to distinguish from each other, and significantly lower WSupCR indicates that they are often also confused with non-cat classes. The performance of all models in terms of WCA, WCP, W10CR, W100CR, and W2CA is concerningly poor, at least at first glance. However, examining the most severe failure cases in detail, we find that many of them can be attributed to problems with the ImageNet-2012 dataset rather than the models. This comes down to clear problems of the ImageNet-2012 validation dataset with label errors in addition to semantically overlapping or completely identical classes, as has for example been discussed by Beyer et al. (2020) and Northcutt et al. (2021). For example, the validation set for "tiger cat" includes several tigers, despite "tiger" being a separate class. Another peculiarity is that there are two classes "maillot" (classes #638 and #639) in the dataset, which unsurprisingly leads to the bad performance of one of the classifiers on this pair (VGG16 using W2CA).

Because of this, we also use the cleaned labels from (Northcutt et al., 2021) and report the same evaluation in Tab. 2. They flagged potentially mislabeled images with suggested new labels for human review. All 3956 out of 50000 samples that humans did not unanimously and uniquely put into either their original or the suggested class are removed in our evaluation. Note that this means that the test set is not completely balanced anymore.

Even after cleaning the labels, the results remain largely the same which might indicate that either the problem of label noise is already in the train set, and thus classifiers have overfitted on the label errors as noted in (Northcutt et al., 2021), or that the given cleaning method is not sufficient.

Table 1. **Evaluation on the full ImageNet-2012 validation set:** using 50000 samples for models indicated in columns. Where applicable, the worst performing classes or superclasses are denoted.

MODEL	RESNET-18	RESNET-50	VGG16	VIT
A	69.76	76.13	71.59	**84.53**
WCA	08.00 (SUNGLASS)	18.00 (MAILLOT)	10.00 (VELVET)	**24.00** (TIGER CAT)
WCP	17.14 (LAPTOP)	23.91 (LAPTOP)	24.39 (LAPTOP)	**30.30** (NOTEBOOK)
WSUPCA	73.20 (CAT)	75.60 (TURTLE)	72.40 (CAT)	**77.60** (CAT)
WSUPCR	60.80 (CAT)	65.60 (CAT)	59.60 (CAT)	**68.80** (CAT)
W10CR	15.40	22.60	18.00	**31.40**
W100CR	34.10	41.92	36.14	**54.16**
W2CA	42.00 (LAPTOP, NOTEBOOK)	44.00 (LAPTOP, NOTEBOOK)	**46.00** (MAILLOT, MAILLOT)	40.00 (LAPTOP, NOTEBOOK)
A@5	89.08	92.86	90.38	**97.29**
WCA@5	38.00 (VELVET)	52.00 (SPOTLIGHT)	36.00 (VELVET)	**68.00** (LETTER OPENER)
W10CR@5	49.40	63.20	52.00	**80.60**
W100CR@5	68.10	77.44	70.22	**89.48**

Table 2. **Evaluation on the cleaned ImageNet-2012 validation set**: using 46044 samples with unambiguous labels from Northcutt et al. (2021) for models indicated in columns. Where applicable, the worst performing classes or superclasses are denoted.

MODEL	RESNET-18	RESNET-50	VGG16	VIT
A	69.79	76.16	71.62	**84.62**
WCA	06.52 (SUNGLASS)	18.75 (MAILLOT)	09.09 (LETTER OPENER)	**21.74** (TIGER CAT)
WCP	15.38 (LADLE)	25.00 (LAPTOP)	26.32 (LAPTOP)	**30.00** (NOTEBOOK)
WSUPCA	71.81 (CAT)	75.00 (TURTLE)	70.93 (CAT)	**76.65** (CAT)
WSUPCR	59.47 (CAT)	63.44 (CAT)	57.71 (CAT)	**66.96** (CAT)
W10CR	13.97	22.32	17.47	**31.73**
W100CR	34.15	41.85	35.81	**54.26**
W2CA	44.18 (LAPTOP, NOTEBOOK)	44.13 (LAPTOP, NOTEBOOK)	**45.26** (MAILLOT, MAILLOT)	40.75 (LAPTOP, NOTEBOOK)
A@5	89.13	92.90	90.42	**97.33**
WCA@5	39.58 (VELVET)	53.19 (SPOTLIGHT)	35.42 (VELVET)	**65.91** (LETTER OPENER)
W10CR@5	49.12	62.06	50.33	**80.09**
W100CR@5	68.16	77.23	70.32	**89.53**

The included metrics which use variations of the top-5 accuracy are an orthogonal remedy to the effect of overlapping classes and some types of mislabeling. We observe that also in this easier and more data sensible setting, for all models, top-5 recall drops significantly when restricting ones view on the worst 10% (W100CR@5), 1% (W10CR@5) or individual (WCA@5) classes.

4. Conclusions and Limitations

We have motivated and proposed several performance measures that help discover class-specific biases both in the classifier and data. As the authors of Balestriero et al. (2022) demonstrated, degradation of performance in some classes can be related to the degradation in performance on a downstream task's class as well. Thus, we conjecture that our measure WSupCA better correlates with the performance on a downstream task related to the worst superclass than clean accuracy would. To test this, we propose evaluating our measures alongside the performance on various downstream tasks on many models. Also, our metrics are not bounded to the specific standard validation set but can be applied to outputs on test datasets that contain shifts or are cleaned in different ways.

A clear limitation of scope is that we only focus on the worst-case classes. For example, Balestriero et al. (2022) showed how some data augmentations can trade-off individual class accuracies for better standard accuracy. If a certain augmentation degrades some class significantly, but not as much compared to any of the worst-n classes, it might not be detected.

ACKNOWLEDGMENTS

The authors acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A) and from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC number 2064/1, Project number 390727645), as well as from the DFG TRR 248 (Project number 389792660). The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Alexander Meinke.

References

Balestriero, R., Bottou, L., and LeCun, Y. The effects of regularization and data augmentation are class dependent. *arXiv:2204.03632*, 2022.

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., and Zhang, Y. "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias", 2018.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with imagenet? *arXiv:2006.07159*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

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

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Maity, S., Xue, S., Yurochkin, M., and Sun, Y. Statistical inference for individual fairness. In *International Conference on Learning Representations*, 2021.

Miller, J. P., Taori, R., Raghunathan, A., Sagawa, S., Koh, P. W., Shankar, V., Liang, P., Carmon, Y., and Schmidt, L. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *PMLR*, pp. 7721–7735, 2021.

Mukherjee, D., Yurochkin, M., Banerjee, M., and Sun, Y. Two simple ways to learn individual fairness metrics from data. In *ICML*, 2020.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. In *NeurIPS*, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *IJCV*, 2015. License: No license specified.

Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Image synthesis with a single (robust) classifier. *Advances in Neural Information Processing Systems*, 32, 2019.

Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., and Schmidt, L. Evaluating machine accuracy on imagenet. In *ICML*, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

Speicher, T., Heidari, H., Grgic-Hlaca, N., Gummadi, K. P., Singla, A., Weller, A., and Zafar, M. B. A unified approach to quantifying algorithmic unfairness: Measuring individual and group unfairness via inequality indices. In *ACM SIGKDD*, 2018.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. *NeurIPS*, 2020.

Tsipras, D., Santurkar, S., Engstrom, L., Ilyas, A., and Madry, A. From imagenet to image classification: Contextualizing progress on benchmarks. In *ICML*, 2020.

Vasudevan, V., Caine, B., Gontijo-Lopes, R., Fridovich-Keil, S., and Roelofs, R. When does dough become a bagel? analyzing the remaining mistakes on imagenet. *arXiv preprint arXiv:2205.04596*, 2022.

Wen, S., Rios, A. S., Lekkala, K., and Itti, L. What can we learn from misclassified imagenet images? *arXiv:2201.08098*, 2022.

Wightman, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Xue, S., Yurochkin, M., and Sun, Y. Auditing ml models for individual bias and unfairness. *AISTATS*, 2020.