Focus on the Positives: Self-Supervised Learning for Biodiversity Monitoring - Supplementary Material

Omiros Pantazis[1] Gabriel J. Brostow[1,2] Kate E. Jones[1] Oisin Mac Aodha[3]
[1]University College London [2]Niantic [3]University of Edinburgh
www.github.com/omipan/camera_traps_self_supervised

1. Additional Experiments

Here we provide additional experiments that highlight the success and failure cases of the different self-supervised learning (SSL) algorithms and our positive image selection mechanisms we evaluated in the main paper.

1.1. Are modern SSL algorithms effective on camera trap data?

From Figure 5 in the main paper we can see that self-supervised methods are superior to ImageNet derived features, in the vast majority of cases. In addition, context-based approaches for positive image sampling further increase the performance. In Figure 2, we can see that while SimCLR is typically the best performing method, there are multiple instances where more naive SSL methods outperform it. Using context-based sampling increases overall accuracy (bottom *vs*. top row), making the simpler baselines competitive with conventional SimCLR.

1.2. Do we need ground truth bounding boxes for SSL to be effective for camera trap data?

In the main paper, we performed all experiments using ground truth bounding boxes. This was to avoid drawing conclusions based on any biases that may be present in a specific detector e.g. some of the datasets we use are public and thus could have been part of the training set for a public detector. To evaluate how effective detected boxes are, we replaced ground truth detections with automatically derived ones from MegaDetector (MD) [1] and retrained our SSL models from scratch on the MMCT dataset. We used MMCT because we can guarantee that MegaDetector is not trained on it. To ensure a fair comparison with the results in the main paper, when performing linear evaluation (but not when performing SSL), we used the ground truth boxes for the test set and supervised training set during evaluation. We can see from the results in Table 2 that the quality of representations from detected boxes are comparable to the ones learned from ground truth (GT) boxes.

Next, we increased the number of detected boxes by a factor of two by adding more images to the training set i.e. from the same camera locations, but at different points in time (denoted as MDx2 v1). This resulted in about 25,406 total cropped images, as opposed to the 13,549 derived from manual annotations. MegaDetector needs no changes or special training to do this. We see that the performance increases compared to using fewer detections. Finally, we tried an alternative version of the above experiment where we added detections from the held out locations in the test set (denoted as MDx2 v2), for the same total of 25,406 cropped images. Again we see an additional improvement but without increasing further the number of images, indicating that the downstream task performance can benefit from pretext training with images from similar locations with the test set.

We conclude that the SSL methods evaluated are robust to how the training cropped images are generated and result in performance that is comparable to using manually annotated boxes. An obvious question is what would the performance be like if you only used entire images and not ones cropped around the objects of interest. Existing work has shown that cropped images are much more effective in the case of image classification from camera traps [2]. Given this and the availability of highly accurate detectors, we chose not to address this question.

1.3. What is the impact of increasing model capacity?

All experiments in the main paper were conducted using a ResNet18 [5] as the backbone feature extractor with an image resolution of 112×112 pixels. As we use images cropped around the objects of interest as input, this resolution is more than sufficient for capturing the visually important characteristics of the categories.

In Table 3 we show the impact of increasing both the backbone model capacity (i.e. using a ResNet50) and the image resolution (i.e. using an image side of size 224). As expected, we see performance improvements across all conditions, but importantly the ranking of the methods is relatively stable. We conclude that for best possible performance, not surprisingly one should use large models and higher resolution. However, this comes at the expense of increased training times and memory consumption.

1.4. What is the impact of initializing end-to-end supervised training with weights learned from self-supervision?

The results in Figure 5 of the main paper were computed with the linear evaluation scheme that is commonly used in SSL i.e. by training a linear model using self-supervised features as input. Here, we use the SSL models as an initialization and fine-tune all the weights of the backbone network to understand if the improvements reported in the linear evaluation case follow through to the end-to-end one. The most realistic setting for camera-trap data is the low label regime where only a small number of images have been annotated. With this in mind, we only present results for the 1% and 10% labeled images settings. When performing supervised training, all hyper parameters are the same as the ones we use for SSL with the exception of the learning rate which we decrease to 0.003. In Table 4 we observe that in almost all cases, models that are initialized with SSL derived weights are vastly superior to those that use only ImageNet initialization (denoted as Standard). Again, we see that the models trained with context result in a much better initialization than without.

1.5. What happens if we start SSL from random weight initialization?

All the results in the main paper use models that have been pretrained on ImageNet. While it is more common in SSL to start from randomly initialized weights, we instead adopted the more pragmatic viewpoint that pretrained networks are readily available, and thus practitioners are likely to use them as a starting point. In the interest of making progress on camera trap image classification (where lack of image labels is the main issue), we chose to start from ImageNet pretrained models. Not surprisingly, if we initialize our SSL models with random weights, the performance is worse on the medium-sized datasets that we use for our experiments (see Table 5). Importantly, we still observe that utilizing context information is superior to standard augmentation-based SSL.

1.6. Are accuracy gains concentrated with the majority-categories, or spread across multiple categories?

In Figure 1 we compare the per-category, 10% linear evaluation, accuracy of standard SimCLR and SimCLR with our context-based positive based sampling. For each category, we also include the number of examples in the 10% subset of the training data. There is no dominant pattern, and we see that context-based sampling helps for both well-represented and under-represented categories.

1.7. What is the proposed algorithm getting right, that "Standard" SSL is getting wrong?

In order to attempt to address this question, we implemented a simple nearest neighbor retrieval visualization. A given query image is used to retrieve the five nearest neighbors in embedding space. In Figure 3 we show example results on CCT20. We can see that the standard augmentation-based SimCLR model leads to retrieved images that look qualitatively like the query in that they have similar lighting and orientation (e.g. portrait *vs*. landscape). But they sometimes contain the wrong animal species, compared to ours (with 'Context'), which seems to capture more diverse appearances and better emphasize species-related characteristics. We observe similar trends for MMCT in Figure 4.

The top-left example in Figure 3 and the top two examples in Figure 4 illustrate a limitation of both of the self-supervised methods. We can see that the oracle is capable of retrieving images with large illumination changes (i.e. spanning night and day). In these examples, it appears that the self-supervised methods are not able to merge these distinct visual modes. This is despite the large amount of color augmenting that they are exposed to during training e.g. color jittering and grayscale conversion. An interesting future question, is what additional information can we make use of during training to merge these diverse modes within a given category.

1.8. Is the proposed approach applicable beyond camera trap data?

While we believe that the four quite distinct camera trap datasets explored in the paper constitute an important problem that deserves dedicated attention, we explore the applicability of our approach on the Functional Map of the World (FMoW) [4] to test the generalizability of the findings. FMoW is a satellite imagery dataset that contains annotated images of categories relevant to the functional purpose of buildings or land use. The data comes in temporal sequences and is accompanied by metadata which make them suitable for validating our approach. We used a subset of the data that consists of 30 different classes, with 30,014 images reserved for training and 10,085 for testing. The results in Table 1, show that: (i) SSL is superior to ImageNet features and (ii) our context selection is consistently better than standard SSL, especially in the low-data regime.

2. Implementation Details

Here we provide additional implementation details related to our experimental evaluation.

2.1. Training

Unless otherwise stated, each SSL network uses a backbone initialized with ImageNet weights. We train all models

for 200 epochs with a batch size of 256 and a learning rate of 0.03, using a cosine annealing schedule. We use SGD with momentum of 0.9 and weight decay of 0.0005. The projector g is a two layer MLP with a hidden layer of size 512 and size 128 for the output. The predictor h, used by SimSiam, is also a two layer MLP with a hidden layer of size 64. For SimSiam only, as in the original paper [3], we add batch normalization to the output of the first layer for the projector and predictor – the model performed poorly without it. For the larger capacity models, we reduced the batch size by half and scaled the learning rate by half also.

For the triplet loss in Equation 1 we set the margin to 0.3. To scale the distances used by SimCLR, we use a temperature of 0.5, see Equation 2 in the main paper. The context temperature parameter in Equation 5 is set to 0.05.

For the sequence positive approach in Section 3.2, we consider all images that are from the same location that are captured within 5 seconds of each other as potential positives.

We use the same set of augmentations for all SSL and end-to-end supervised methods at training time. This includes random resized crops in the range $[0.2, 1.0]$, horizontal flipping with probability 0.5, color jittering with probability 0.8, and grayscale conversion with probability 0.2.

2.2. Evaluation

When training the linear classifiers for evaluation we use logistic regression with L2 regularization as implemented in `scikit-learn` [1]. We use a lbfgs solver, with a maximum number of iterations of 1000 and a multinomial loss. To select the regularization weighting, we search over the set $\{0.001, 0.01, 0.1, 1, 10, 100\}$, and choose the best value using five-fold cross validation on the training set. When training the linear classifier, we simply resize the cropped image to the desired image resolution (e.g. 112×112) when extracting features, and we do not use any other augmentation.

When generating the 1% and 10% subsets, we randomly sampled the corresponding percentage of images of each class in the full training set, while also ensuring that there were at least one example per class. Some categories are more common than others, so this sub-sampling procedure preserves the imbalance that is typical in camera trap datasets. We use the same fixed subsets for all experiments for a given dataset.

Functional Map of the World (FMoW)				
Approach	Method	1%	10%	100%
Supervised End-to-End*	-	37.62	51.79	58.16
Supervised	-	36.23	46.24	52.72
Triplet	Standard	44.38	54.05	54.58
	Seq. Pos.	**50.66**	**55.22**	56.67
	Con. Pos.	49.75	55.16	**57.28**
SimCLR	Standard	45.60	51.53	52.16
	Seq. Pos.	45.30	50.07	50.85
	Con. Pos.	**45.93**	**52.24**	**53.03**
SimSiam	Standard	46.58	51.74	**55.76**
	Seq. Pos.	46.63	51.50	52.97
	Con. Pos.	**47.64**	**53.51**	52.80

Table 1. Linear evaluation of our approach on a subset of FMoW, a satellite imagery dataset. Comparison of SSL approaches shows that using context benefits in most cases, especially in the low-data regime. The results here are averaged over three runs. *We also include fully supervised end-to-end training for reference.

References

[1] Sara Beery, Dan Morris, and Siyu Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019. 1

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018. 1

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[4] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *CVPR*, 2018. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[1] scikit-learn: https://scikit-learn.org

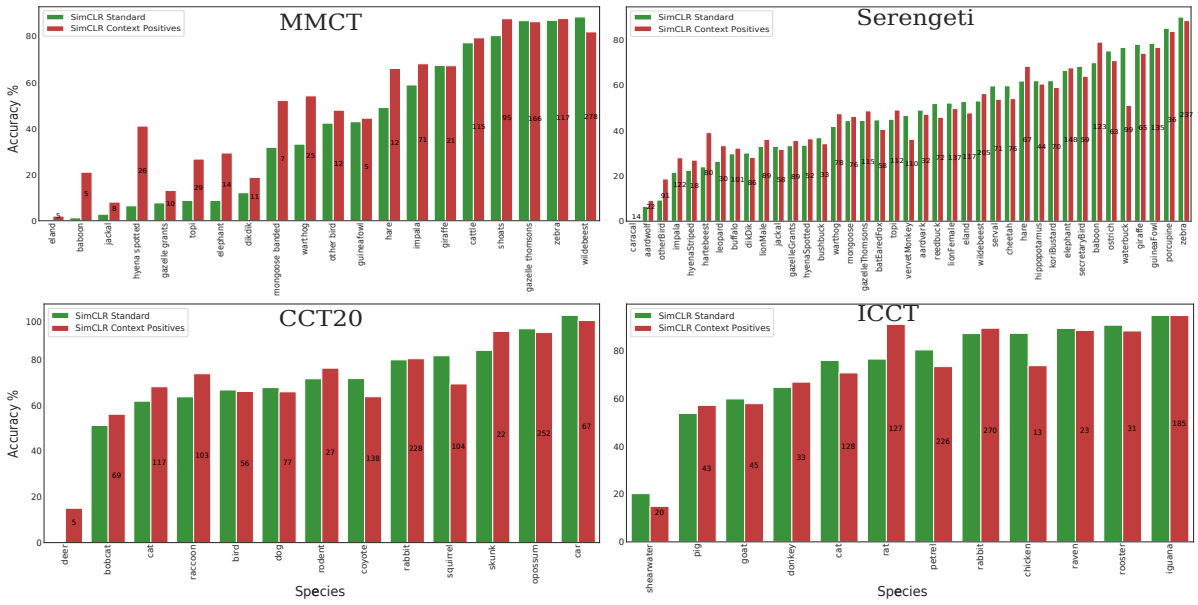

Figure 1. Evaluating per-class accuracy. Here we report test-time, per-class, linear evaluation accuracy for each of our four different datasets. We compare standard image augmentation to our context-based approach using SimCLR as the SSL algorithm.

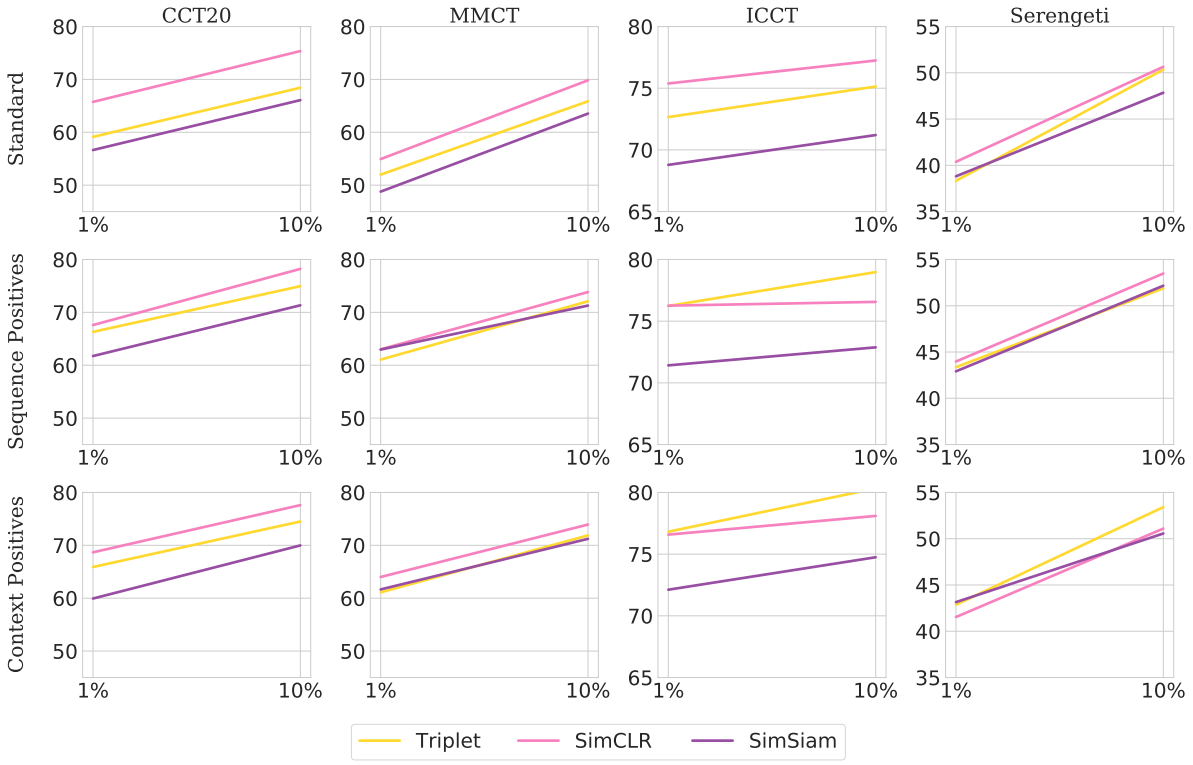

Figure 2. Linear evaluation accuracy of different SSL approaches and positive pair selection strategies. This is an alternative presentation of the results from Figure 5 from the main paper where it is easier to compare the different SSL methods on a given dataset.

Maasai Mara Camera Traps (MMCT)									
		1%				10%			
Approach	Method	GT	MD	MDx2 v1	MDx2 v2	GT	MD	MDx2 v1	MDx2 v2
Triplet	Standard	51.99	52.93	55.35	55.76	65.86	64.85	68.47	68.43
	Seq. Pos.	61.06	60.63	**61.63**	62.82	**72.09**	70.79	72.27	73.25
	Con. Pos.	**61.10**	**60.79**	61.51	**63.52**	71.88	**70.88**	**72.66**	**73.27**
SimCLR	Standard	54.93	56.30	59.46	60.32	69.83	68.95	72.67	73.46
	Seq. Pos.	63.00	62.55	**64.11**	64.57	73.84	72.21	**74.37**	74.37
	Con. Pos.	**64.01**	**63.84**	63.98	**67.29**	**73.93**	**72.77**	74.03	**76.01**
SimSiam	Standard	48.78	47.87	56.75	57.17	63.53	62.98	68.33	68.11
	Seq. Pos.	**62.96**	61.18	63.08	**66.90**	**71.28**	70.46	72.29	**74.73**
	Con. Pos.	61.66	**62.78**	**64.83**	66.59	71.23	**70.78**	**72.40**	74.01

Table 2. Comparing detected bounding boxes from MegaDetector (MD) with ground truth (GT) boxes on the MMCT dataset. We also evaluate the effect of doubling the camera trap detections from locations that are not in the test set (MDx2 v1) and locations that belong to the test set (MDx2 v2).

Caltech Camera Traps (CCT20)							
		1%		10%		100%	
Approach	Method	RN18-112	RN50-224	RN18-112	RN50-224	RN18-112	RN50-224
Triplet	Standard	59.13	61.99	68.43	73.85	73.40	80.10
	Seq. Pos.	**66.30**	70.80	**74.96**	80.96	**77.31**	**83.67**
	Con. Pos.	65.90	**70.87**	74.51	**81.22**	76.42	83.11
SimCLR	Standard	65.75	68.59	75.36	78.98	76.37	83.39
	Seq. Pos.	67.60	74.96	**78.22**	82.74	**78.99**	**85.54**
	Con. Pos.	**68.67**	**75.06**	77.60	**82.78**	78.02	85.14
SimSiam	Standard	56.64	57.47	66.08	72.42	72.13	78.61
	Seq. Pos.	**61.74**	66.16	**71.34**	78.39	**74.78**	81.52
	Con. Pos.	59.93	**69.95**	69.98	**79.14**	74.38	**82.59**

Table 3. Evaluating model capacity. Linear evaluation accuracy on CCT20 with increased model capacity from ResNet18 (RN18) to ResNet50 (RN50) and image resolution from (112×112) to (224×224). We compare the positive-pair mining methods across all SSL approaches.

			1%		10%	
Dataset	Approach	Method	Lin. Eval.	End-to-End	Lin. Eval	End-to-End
CCT20	Supervised	-	-	55.65	-	75.14
	SimCLR	Standard	65.75	68.76	75.36	78.20
	SimCLR	Seq. Pos.	67.60	**72.56**	**78.21**	**79.33**
	SimCLR	Con. Pos.	**68.67**	72.06	77.61	78.73
MMCT	Supervised	-	-	53.00	-	67.73
	SimCLR	Standard	54.93	59.30	69.82	73.97
	SimCLR	Seq. Pos.	63.00	**62.91**	73.84	74.22
	SimCLR	Con. Pos.	**64.00**	62.23	**73.93**	**75.89**
ICCT	Supervised	-	-	62.50	-	76.60
	SimCLR	Standard	75.38	75.14	77.24	**79.44**
	SimCLR	Seq. Pos.	76.26	76.02	76.56	77.38
	SimCLR	Con. Pos.	**76.58**	**77.13**	**78.10**	79.12
Serengeti	Supervised	-	-	36.03	-	**55.29**
	SimCLR	Standard	40.37	41.30	50.64	54.17
	SimCLR	Seq. Pos.	**43.97**	**43.79**	**53.48**	54.58
	SimCLR	Con. Pos.	41.53	42.62	51.10	54.19

Table 4. Comparing linear versus end-to-end supervised finetuning. Starting from SimCLR derived self-supervised representations, we compare linear evaluation (as in the main paper) to end-to-end supervised finetuning from SSL initialization. 'Supervised' refers to the performance of the fully-supervised baseline, initialized from ImageNet only, without using SSL.

Caltech Camera Traps (CCT20)							
		1%		10%		100%	
Approach	Method	ImageNet	Random	ImageNet	Random	ImageNet	Random
Triplet	Standard	59.13	45.22	68.43	51.04	73.40	56.14
	Seq. Pos.	**66.30**	46.53	**74.96**	**55.24**	**77.31**	**59.35**
	Con. Pos.	65.90	**46.90**	74.51	54.65	76.42	59.19
SimCLR	Standard	65.75	53.29	75.36	62.90	76.37	66.89
	Seq. Pos.	67.60	58.09	**78.22**	**67.40**	**78.99**	**70.84**
	Con. Pos.	**68.67**	**58.86**	77.60	67.36	78.02	69.41
SimSiam	Standard	56.64	41.36	66.08	48.77	72.13	53.45
	Seq. Pos.	**61.74**	43.04	**71.34**	50.70	**74.78**	54.92
	Con. Pos.	59.93	**43.82**	69.98	**51.35**	74.38	**56.27**

Table 5. Comparing random initialization to ImageNet initialization. Linear evaluation comparison of SSL approaches on CCT20 with the backbone feature extractor f initialized with weights either from ImageNet (as in the main paper) or randomly.

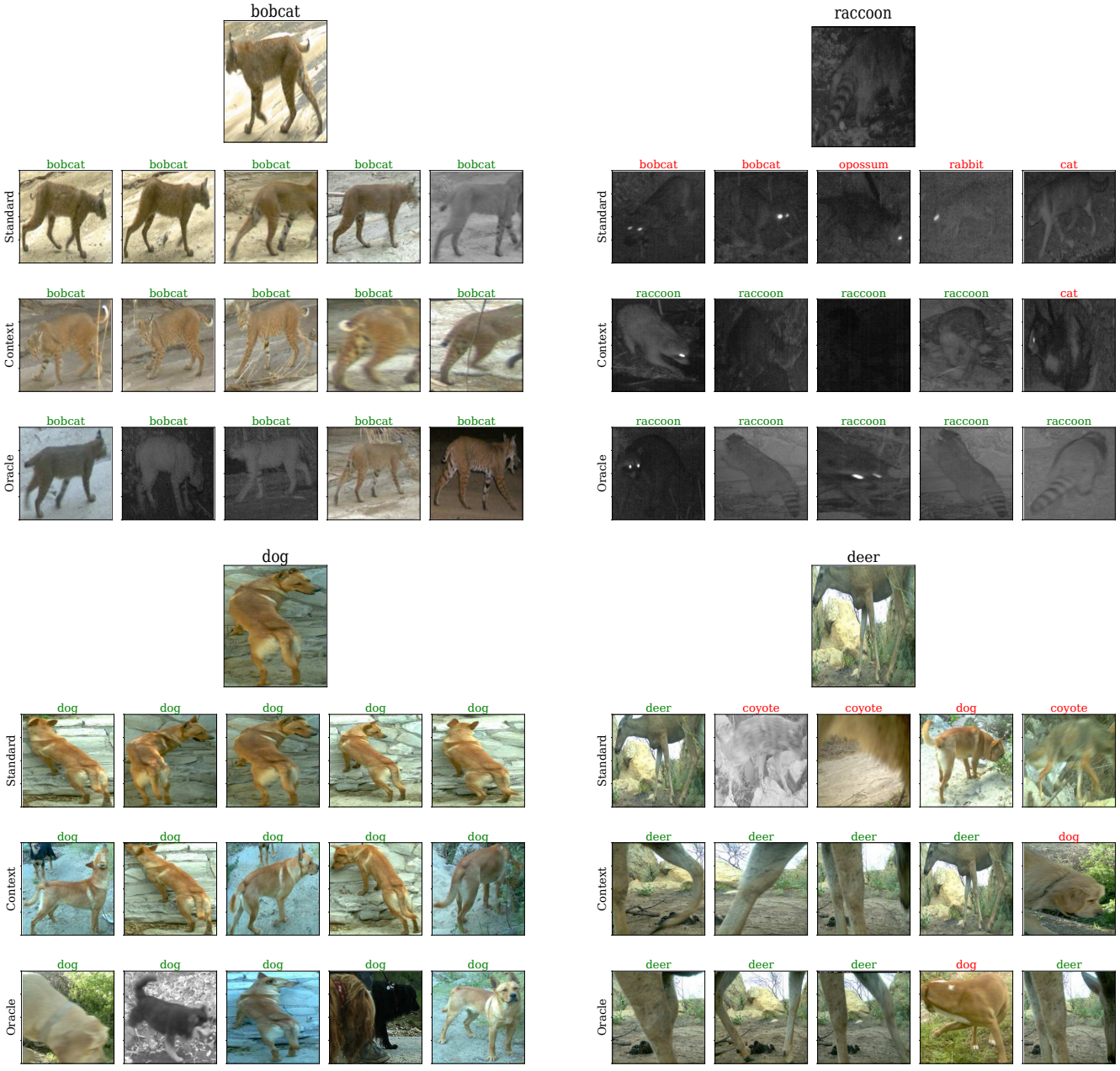

Figure 3. Nearest neighbor retrieval results for SimCLR for four different test images from CCT20. For each of the different images, we show the top five nearest neighbors in 128 dimensional embedding space (i.e. the output of the projector) for three different SimCLR models. The label on top of each image is the ground truth class name. We see that the nearest neighbors for 'Standard' SimCLR display very limited visual diversity. The unobtainable 'Oracle' model, which has been trained with ground truth labels, has the most variety. Our 'Context' approach is between the two extremes and shows non-trivial diversity which indicates that it contains more semantic information in it's features compared to conventional augmentation-based SimCLR. Note, that none of these images have been observed during self-supervised training.

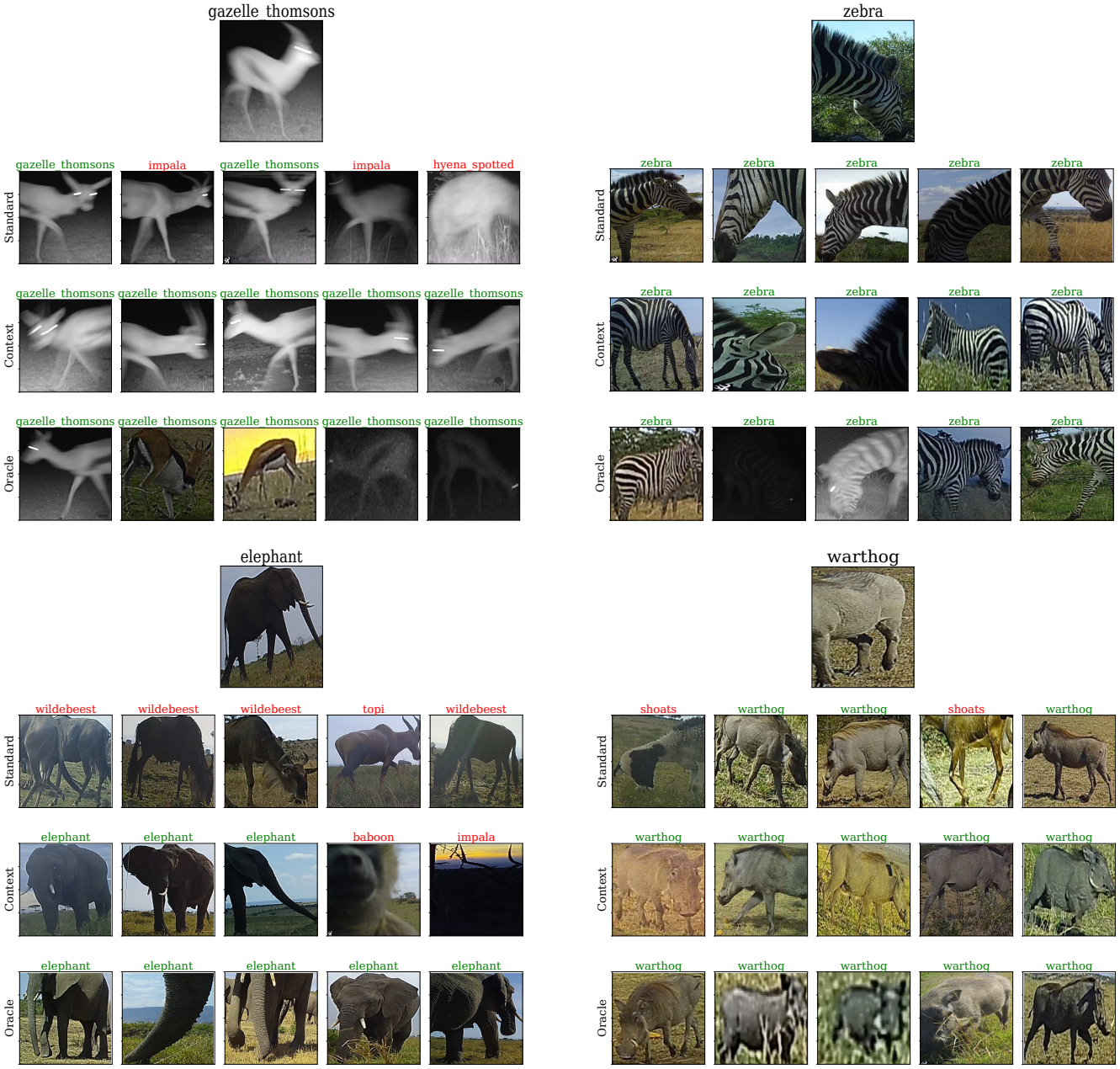

Figure 4. Nearest neighbor retrieval results for SimCLR for four different test images from MMCT. For each of the different images, we show the top five nearest neighbors in 128 dimensional embedding space (i.e. the output of the projector) for three different SimCLR models. The label on top of each image is the ground truth class name. Again, our 'Context' approach captures more diverse appearances and perspectives of animals compared to conventional augmentation-based SimCLR. Note, that none of these images have been observed during self-supervised training.