Supplementary Material for FACTS: First Amplify Correlations and Then Slice to Discover Bias

Sriram Yenamandra Pratik Ramesh Viraj Prabhu Judy Hoffman
Georgia Institute of Technology
{sriramy, pratikramesh, virajp, judy}@gatech.edu

Contents

1. Additional Dataset Details

We provide the counts of number of samples in *train*, *val* and *test* splits for our evaluation settings in Table 1. While we use standard splits for CelebA and Waterbirds proposed in prior bias mitigation work [1], we propose our own splits for NICO++, which we now describe in more detail.

NICO++ splits. To have a sufficient number of samples per-group, we first group concepts (eg. cats, dogs, trains, bikes) to form six super-concepts: *mammals*, *birds*, *plants*, *waterways*, *landways* and *airways*, each occurring in six contexts, giving us a total of $6 \times 6 = 36$ groups. We then create settings where each context is spuriously correlated with a unique super-concept (*e.g. rocks* (context) and *mammals* (super-class)). This results in 6 bias-aligned and 30 bias-conflicting slices. Table 2 describes the dominant contexts for each super-class, alongwith the base NICO++ classes included in the super-class.

To generate splits, we first randomly select 50 images per each (super-concept, context) pair for creating our evaluation *test* split. For each of NICO++^{95}, NICO++^{90}and NICO++^{75}, we then create a *trainval* (*train* + *val*) split.

Setting	*train*	*val*	*test*
Waterbirds	4795	1199	5794
CelebA	162770	19867	19962
NICO++^{75}	9349	2349	1800
NICO++^{90}	8209	2074	1800
NICO++^{95}	7839	1979	1800

Table 1: Number of samples in *train*, *val* and *test* splits of our evaluation settings.

To create bias-aligned slices from the *trainval* split, we first select a super concept and its corresponding dominant concept (*e.g. mammals* and *rocks*) and retrieve all images annotated with both. Next, we select the required number of bias-conflicting samples (where the super concepts occur in *non-dominant* contexts) such that a desired correlation strength of $\beta \in \{75, 90, 95\}$ is ensured for each NICO++$^{\beta}$ setting. Finally, we divide the train-val split uniformly at random in an 80-20 ratio to form the *train* and *val* splits respectively. Table 3 shows the resulting train distribution of classes and contexts for NICO++^{90}.

2. FACTS: Additional analysis

2.1. GradCAM visualizations

We generate GradCAM visualizations [2] to investigate the region of interest of the standard ERM model h_s and the bias-amplified model $h_{\texttt{AmCo}}$ across the bias-conflicting samples. Figure 1 displays some of the samples belonging to the bias-conflicting slices which were correctly classified using the ERM model. We observe that h_s focuses on features associated with the target label, whereas $h_{\texttt{AmCo}}$ focuses on features associated with the spurious attribute. We also observe that $h_{\texttt{AmCo}}$ mispredicts the samples as the class that is correlated with the spurious attribute.

Super-concept	**NICO++ concepts**	**Dominant context**
mammals	sheep, wolf, lion, fox, elephant, kangaroo, cat, rabbit, dog, monkey, squirrel, tiger, giraffe, horse, bear, cow	rock
birds	bird, owl, goose, ostrich	grass
plants	flower, sunflower, cactus	dim lighting
landways	bicycle, motorcycle, train, bus, scooter, truck, car	autumn
waterways	sailboat, ship, lifeboat	water
airways	hot air balloon, airplane, helicopter	outdoor

Table 2: **NICO++ split details**. We list super-concepts that we use as target labels and their corresponding base concepts from the original NICO++ dataset. Each super-concept co-occurs with six different contexts. In our proposed train and validation splits, each context *dominantly* co-occurs with a unique super-concept (*e.g.* *rocks* and *mammals*) .

Contexts \ Classes	mammals	birds	plants	airways	waterways	landways
rock	2552	50	50	50	50	50
grass	24	1280	24	24	24	24
dim lighting	12	12	616	12	12	12
outdoor	16	16	16	879	16	16
water	20	20	20	20	1063	20
autumn	21	21	21	21	21	1104

Table 3: **NICO++ split distribution.** Distribution of samples for each (super-concept, context) pair in the *train* split of NICO++^{90}.

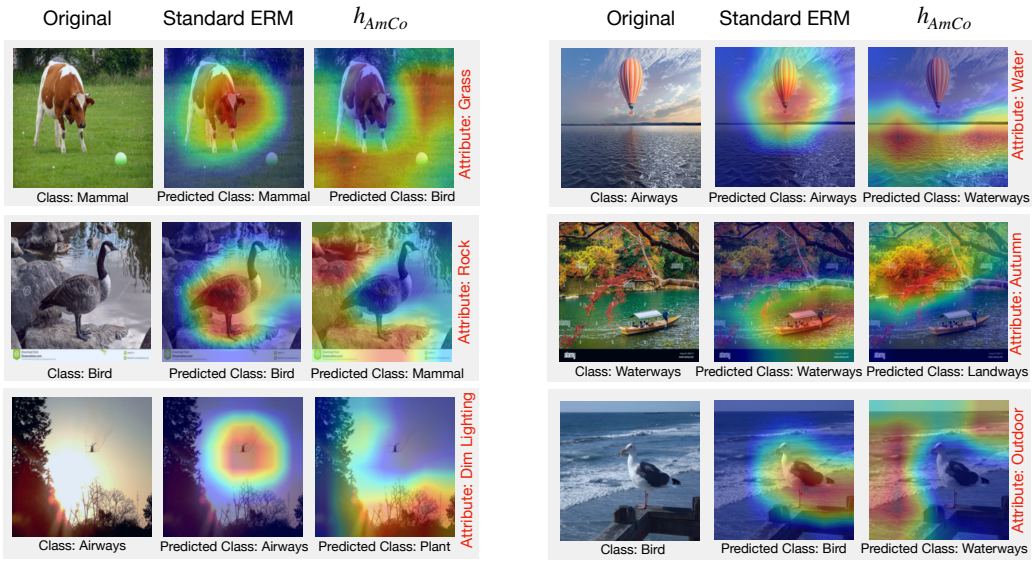

Figure 1: GradCAM [2] visualizations of bias-conflicting samples which were correctly classified by the ERM model. We see that the bias-amplified model $h_{\texttt{AmCo}}$ makes predictions focusing on the features associated with the spurious attribute, whereas the standard ERM model is able to learn some of the features associated with the target label.

2.2. `CoSi` Sensitivity to hyperparameters

For the NICO++^{95} setting, we find $\alpha = 0.25$, $\Sigma_p \cong$ *full* and $\delta_p = 10^{-3}$ to result in the best Silhouette coeffi-

cient [3]. In Table 4, we vary the different hyperparameters used in the second stage of our approach. First, we vary the value of α - the weight assigned to the *correlation prior* in the log-likelihood (Section 3.2). Next, we change the covariance type of Σ_p, the covariance of the mutivariate normal distribution used to model the *correlation prior*. We find that restricting covariance to *diagonal* drops the `Precision@10` to 0.53 from 0.62, while restricting all mixture components to have a *tied* covariance matrix drops the `Precision@10` to 0.58. Finally, fitting the mixture model using the logits from our bias-amplified ERM model $h_{\texttt{AmCo}}$ to a standard ERM model drops the `Precision@10` to 0.36.

Ablation	`Precision@10`
$\alpha = 25$, $\Sigma_p \cong$ *full*, $\delta_p = 10^{-3}$ (ours)	**0.62**
$\alpha = 10$	0.57
$\alpha = 50$	0.53
$\Sigma_p \cong$ *diagonal*	0.53
$\Sigma_p \cong$ *tied*	0.58
$\delta_p = 10^{-5}$	0.57
$\delta_p = 10^{-4}$	**0.62**
$\delta_p = 10^{-2}$	0.49
Using standard ERM	0.36

Table 4: **Ablating `CoSi` hyperparameters.** Results on NICO++^{95}.

2.3. `CoSi` hyperparameter tuning and validation

In Figure 2, we show how the use of Silhouette coefficient [3] results in selection of hyperparameters achieving good `Precision@10` in the NICO++^{95} setting. The Silhouette coefficient measures how well separated different clusters are in the embedding space. We compute the mean of Silhouette coefficients obtained in the embedding spaces of CLIP [4] and model predictions.

2.4. Decaying weight decay schedule

Our current method requires training multiple models with different weight decays for finding the right capacity of model needed. Here, we explore varying the weight decay in a single training run and then picking the model that achieves highest average variation in per-class confidences. Specifically, we decay the weight decay exponentially from 2.0 to 10^{-3} over the course of training. While this simple strategy results in training of far lesser models, we find that this doesn't result in consistent gains in the more difficult NICO++ settings.

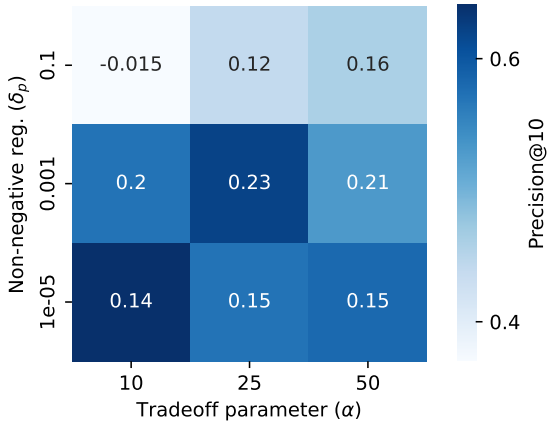

Figure 2: **Validation of `CoSi` hyperparameter selection strategy.** We plot the `Precision@10` (color, darker is higher) as the hyperparameters in `CoSi` (δ_p and α) vary along with the values of Silhouette coefficients (inscribed values). We select $\delta_p = 0.001$ and $\alpha = 25$ that result in the highest value of Silhouette coefficient.

Identification	Waterbirds	CelebA	NICO++^{90}	NICO++^{95}
`AmCo`	0.58	0.29	**0.41**	**0.31**
wd schedule	**0.63**	**0.31**	0.35	0.30

Table 5: Comparison of methods in terms of `Avg-AP` for retrieving bias-conflicting samples across evaluation settings.

2.5. Additional qualitative examples

In Figs. 3-12 we present additional qualitative examples of slices discovered by `FACTS` on the CelebA, Waterbirds, and NICO++ settings. We present the top-6 slices after ranking the slices based on model's performance on the slice. We report model's accuracy on each slice at the top.

3. `FACTS`: Additional comparison to prior work

3.1. Domino

- Our method makes use of a bias-amplified ERM model $h_{\texttt{AmCo}}$ instead of the standard ERM model h_b. This helps increase the separation between the bias-aligned and bias-conflicting samples belonging to a particular class (validated in Section 4.5).

- Our method makes use of richer prior in the form of logits $h_{\texttt{AmCo}}(X)$ produced by the bias-amplified model instead of the categorical assignment of the predicted class label $\hat{Y} = argmax(h_b(X))$ using a standard ERM model.

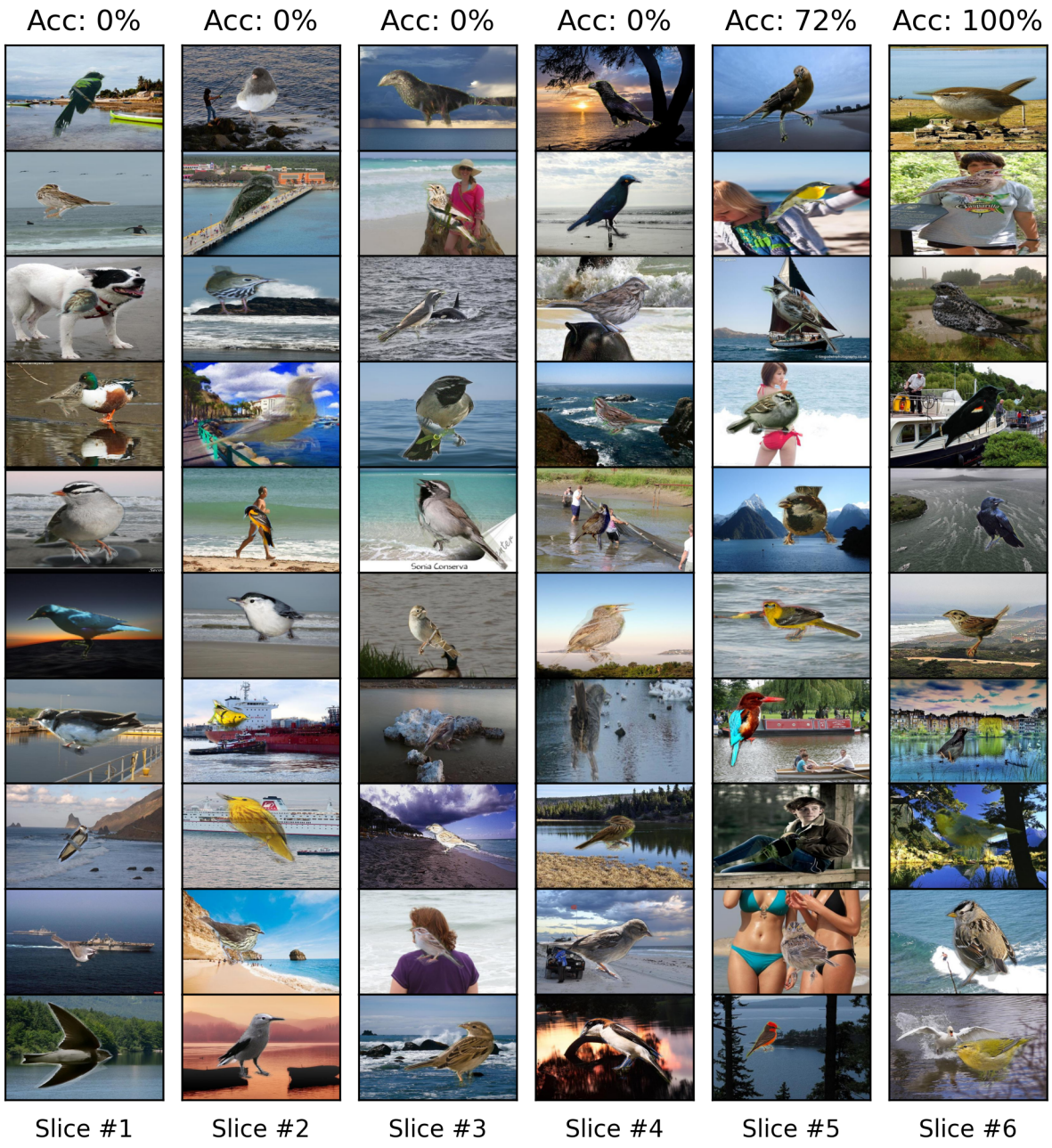

Figure 3: Top-6 slices retrieved by FACTS for the *landbirds* class from Waterbirds. All these slices predominantly contain the bias-conflicting slices: *landbirds* in *water* backgrounds.

- Lastly, [5] clusters samples across all classes together and separates them by enforcing a soft constraint on the class membership. It does this by jointly modelling its mixture model with class information using the term $P(\hat{Y} = h_s(x_i)|S^{(j)} = 1)$. We found enforcing a hard constraint helps prevent any inter-class contamination of slices.

We summarize the differences with Domino [5] in Table 6.

3.2. JTT and GroupDRO

Prior mitigation works [6, 1] use high regularization and low learning rates to achieve good worst group performance. Also, [6] observes high weight decay to increase the accuracy gap between bias-aligned and bias-conflicting

Figure 4: Slices retrieved by FACTS for the *waterbirds* class from Waterbirds. The samples in these discovered slices are predominantly belong to the bias-conflicting slice of *waterbirds* in *land* backgrounds.

Criterion	FACTS	DOMINO
Amplification	High l_2 reg.	Standard l_2 reg.
Prior	Bias amplified logits	Cat. assignment of pred. label
Clustering	Per-class, hard assignment	Global, soft assignment

Table 6: Comparing FACTS to DOMINO.

groups. In this work, we exploit this observation for better separating bias-aligned and bias-conflicting groups for the purpose of identifying spurious correlations.

Figure 5: Slices retrieved by FACTS for the *non-blonde* (dark-haired or gray-haired) class of celebrity faces from CelebA. Note that although this class doesn't have a *bias-conflicting* slice, FACTS is able to recover coherent slices with degraded performance.

4. FACTS: Additional details

4.1. Implementation Details for CoSi

In the second stage of our approach, we first initialize the slices using the model's confusion matrix over validation data following [5], wherein samples with identical predictions are assigned to the same slices. We fit 36 mixture components per-class. Once the mixture model is fit, we assign each sample to the slice under which the sample achieves the highest density, and rank samples within a slice in order of this density. Finally, we rank slices using model performance and report top-6 slices.

For generating captions and keywords for slices (in Fig.

Figure 6: Slices retrieved by FACTS for the *blonde* class from CelebA. Here, FACTS recovers multiple slices (Slice #1, #3, #4) corresponding to the *bias-conflicting* slice: *blonde males*.

5) we closely follow [7]. Specifically, we use an off-the-shelf captioning model, ClipCap [8] for naming all images in a slice. To each slice, we assign a keyword that repeats the most number of times within the captions of the slice.

References

[1] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–

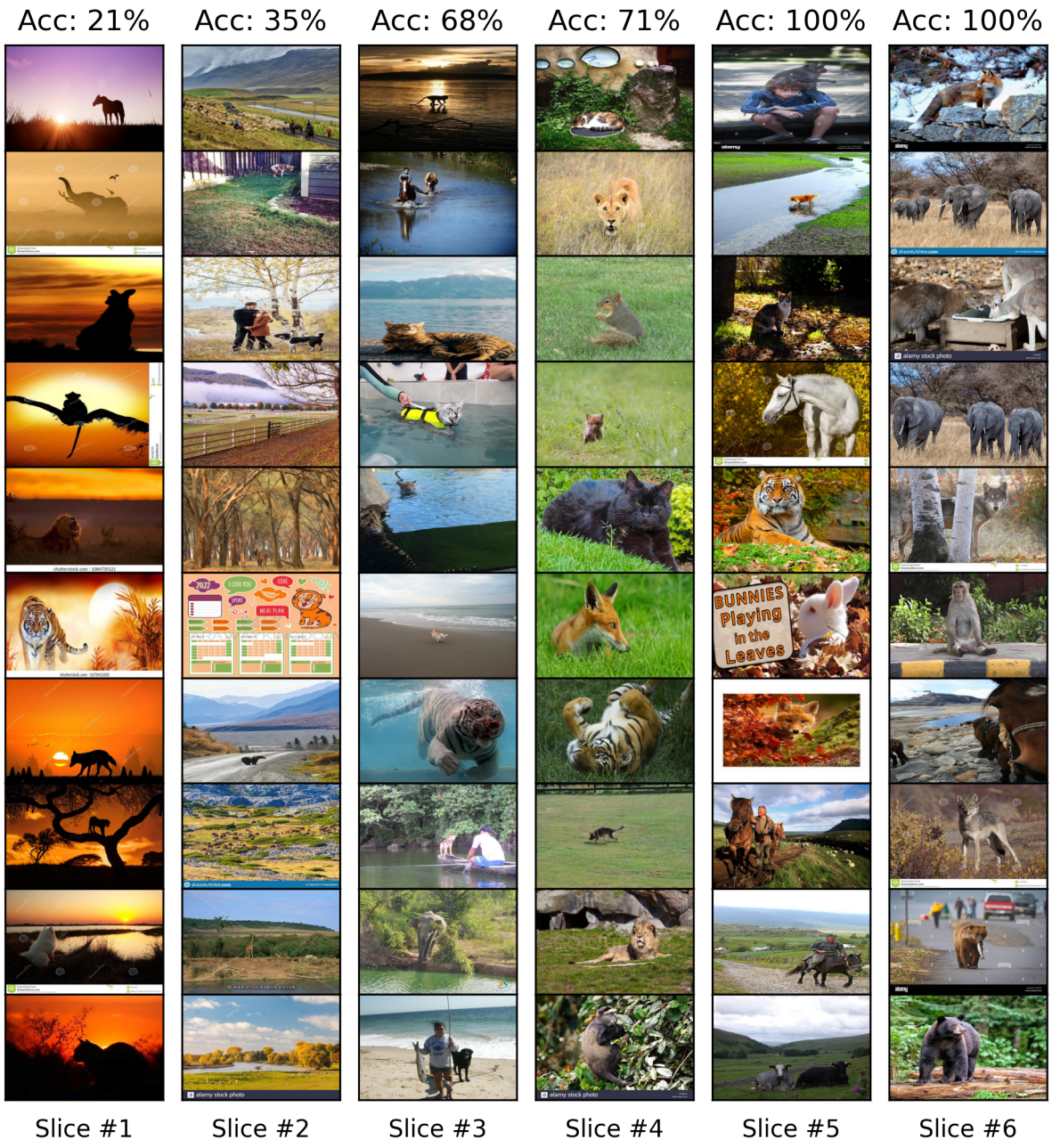

Figure 7: Slices retrieved by FACTS for the *mammals* class from NICO++[90]. Note that the dominant context for *mammals* is *rocks*.

24 Jul 2021. 1, 4

[2] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 2

[3] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 3

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[5] Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-

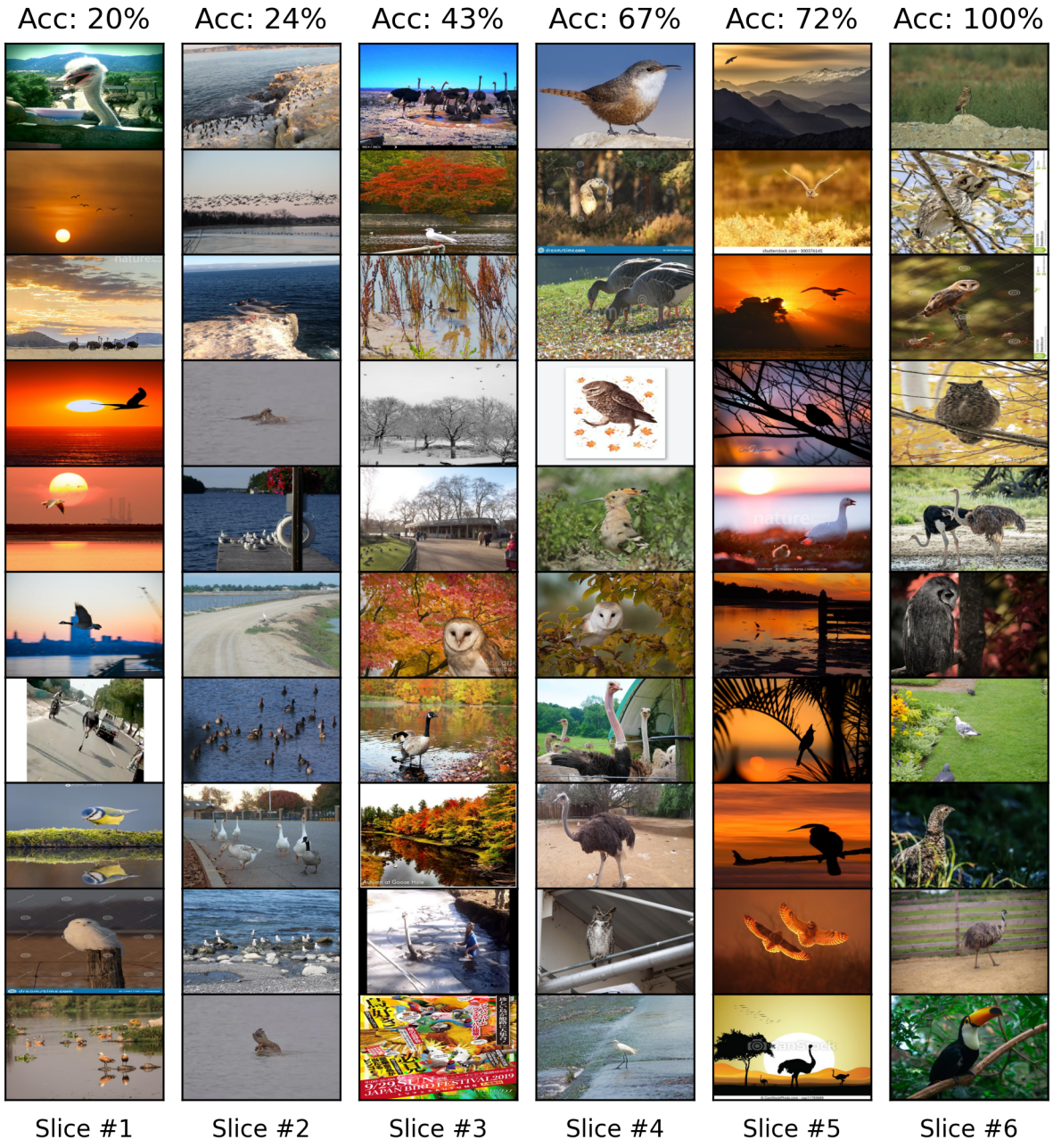

Figure 8: Slices retrieved by FACTS for the *birds* class from NICO++[90]. Note that the dominant context for *birds* is *grass*.

Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 4, 6

[6] Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. 4

[7] Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Explaining visual biases as words by generating captions. *arXiv preprint arXiv:2301.11104*, 2023. 7

[8] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 7

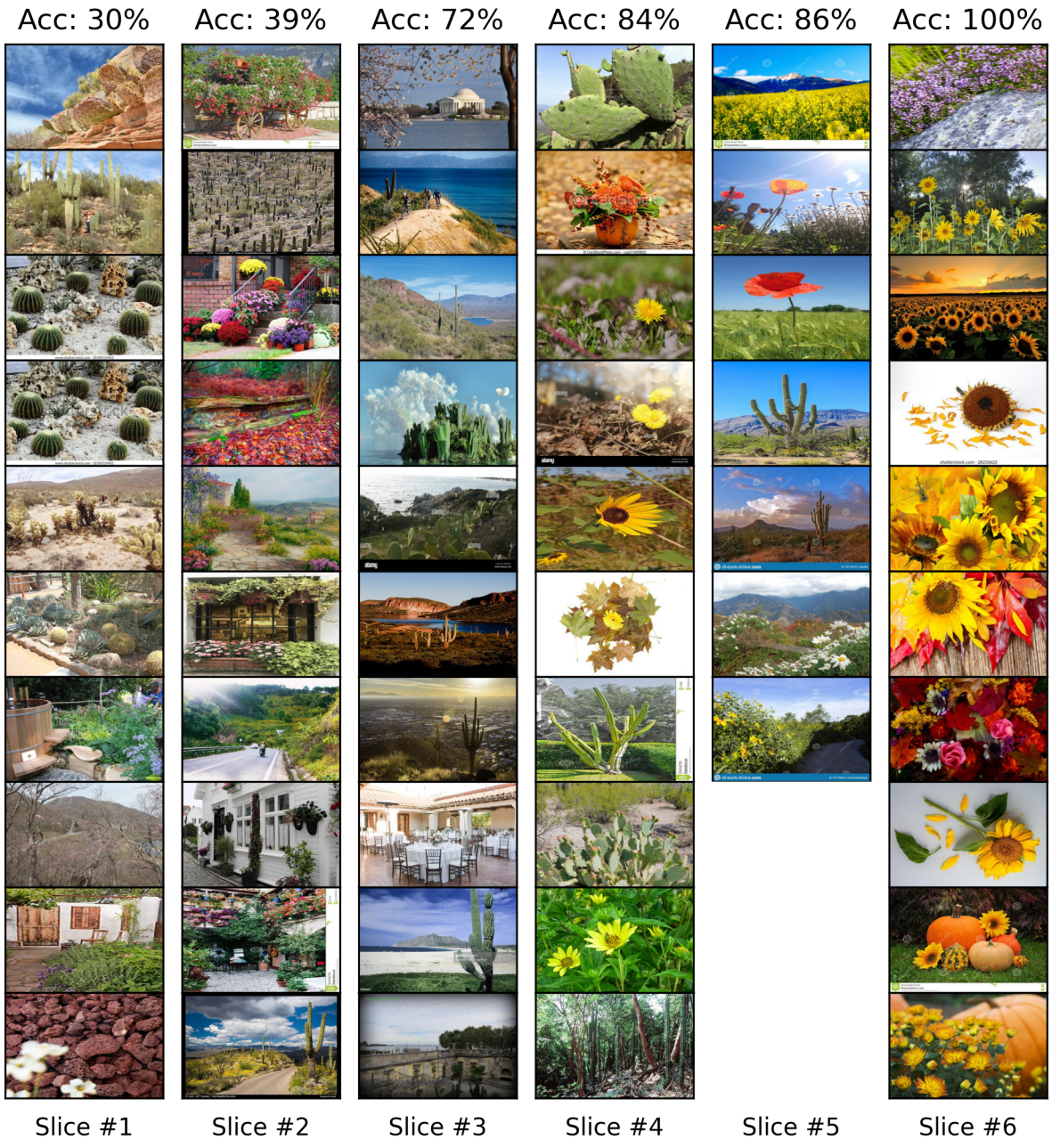

Figure 9: Slices retrieved by FACTS for the *plants* class from NICO++[90]. Note that the dominant context for *plants* is *dim lighting*.

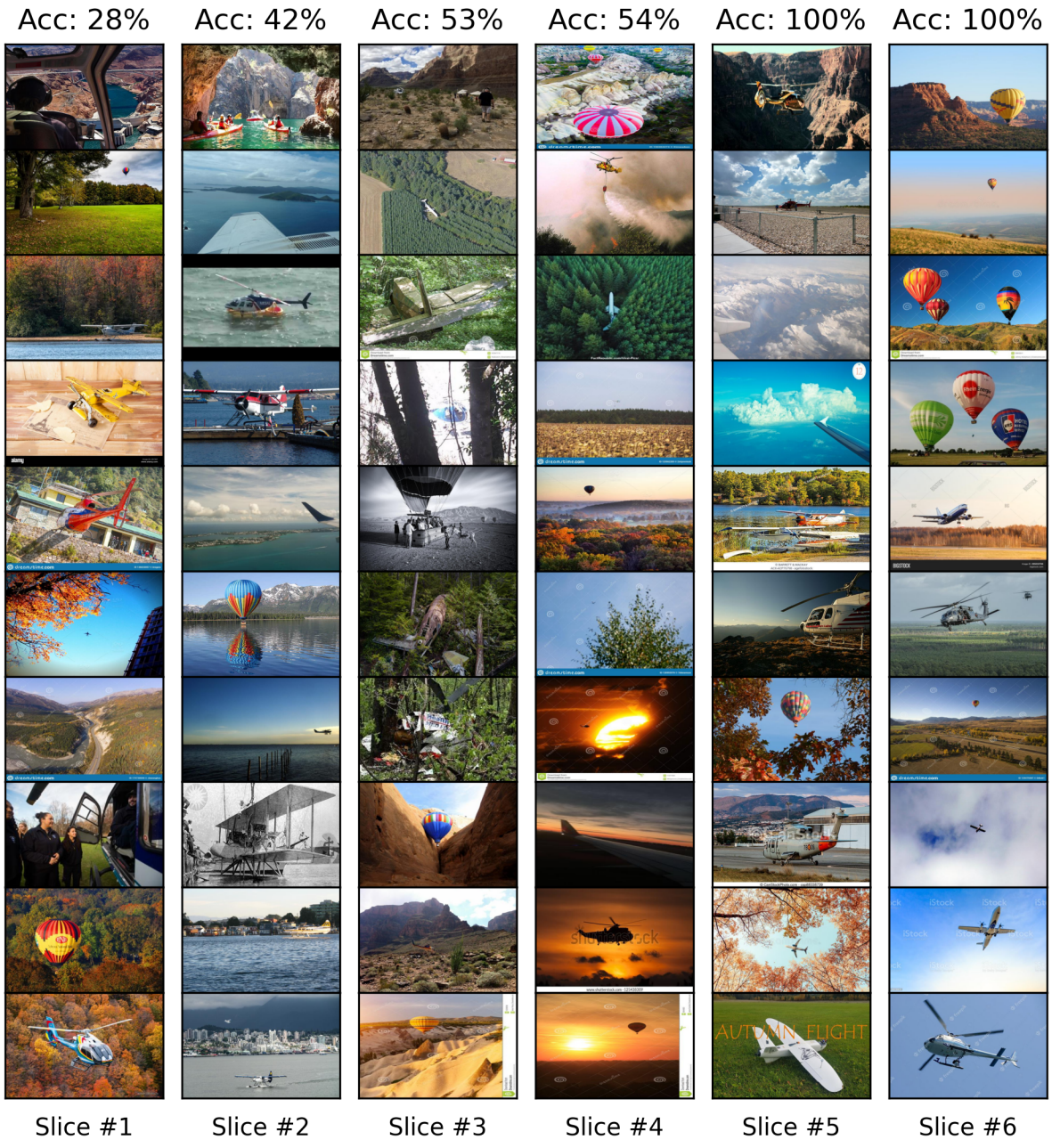

Figure 10: Slices retrieved by FACTS for the *airways* class from NICO++[90]. Note that the dominant context for *airways* is *outdoor*.

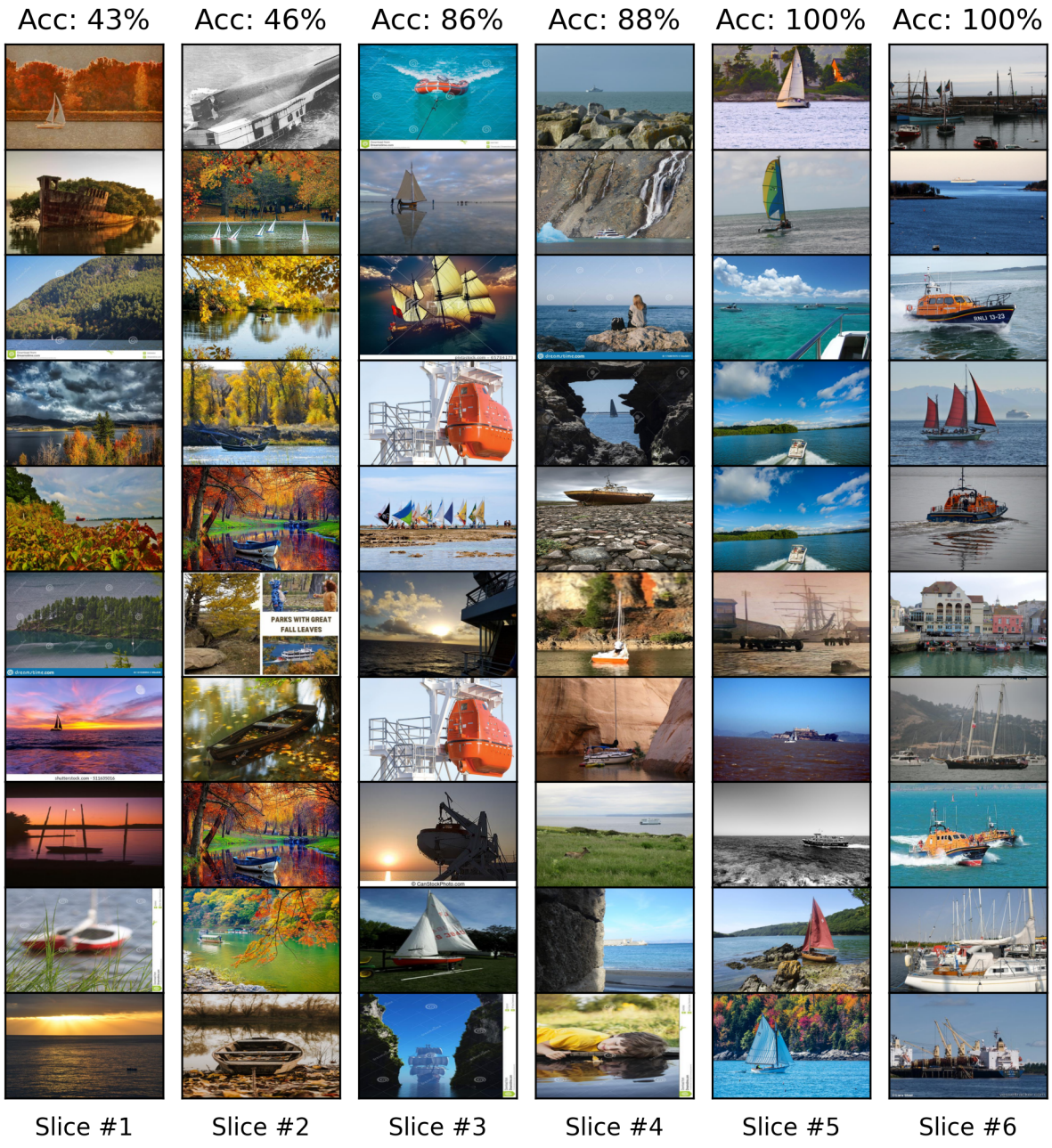

Figure 11: Slices retrieved by FACTS for the *waterways* class from NICO++[90]. Note that the dominant context for *waterways* is *water*.

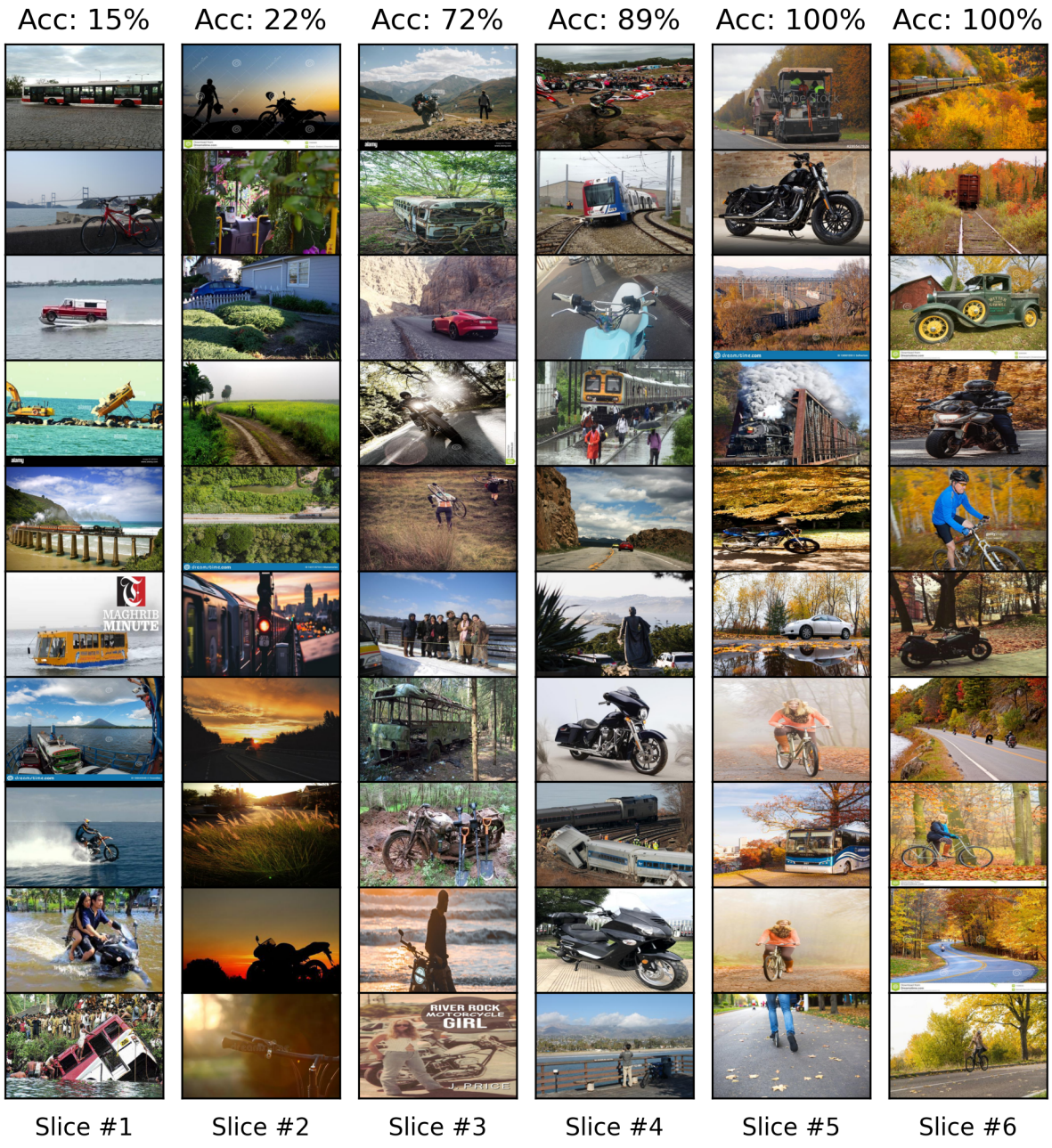

Figure 12: Slices retrieved by FACTS for the *landways* class from NICO++[90]. Note that the dominant context for *landways* is *autumn*.