arXiv:2508.10637v2 [cs.CV] 1 Apr 2026

Processing and acquisition traces in visual encoders: What does CLIP know about your camera?

Ryan Ramos[1*] Vladan Stojnić[2*] Giorgos Kordopatis-Zilos[2] Yuta Nakashima[1]
Giorgos Tolias[2] Noa Garcia[1]
[1]The University of Osaka [2]VRG, FEE, Czech Technical University in Prague

Figure 1. **Impact of metadata on the representation space of visual encoders.** The similarity in the representation space of foundation visual encoders is influenced not only by semantic labels but also by metadata labels. This work explores and uncovers these effects for metadata related to image processing (*e.g.*, JPEG compression) and image acquisition (*e.g.*, camera model). The figure illustrates the search results for the query image with specific metadata labels. Retrieved images are ranked according to their similarity to the query. Different image collections exhibit varied combinations of semantic and metadata labels, affecting the retrieval outcome.

Abstract

Prior work has analyzed the robustness of visual encoders to image transformations and corruptions, particularly in cases where such alterations are not seen during training. When this occurs, they introduce a form of distribution shift at test time, often leading to performance degradation. The primary focus has been on severe corruptions that, when applied aggressively, distort useful signals necessary for accurate semantic predictions.

We take a different perspective by analyzing parameters of the image acquisition process and transformations that may be subtle or even imperceptible to the human eye. We find that such parameters are systematically encoded in the learned visual representations and can be easily recovered. More strikingly, their presence can have a profound impact, either positively or negatively, on semantic predictions. This effect depends on whether there is a strong correlation or anti-correlation between semantic labels and these acquisition-based or processing-based labels. Our code and data are available at: https://github.com/ryan-caesar-ramos/visual-encoder-traces

1. Introduction

Visual encoders are fundamental to computer vision algorithms, serving as the backbone for projecting images into meaningful representations. This core pipeline underpins various recognition tasks, from image classification [21, 35] and retrieval [34, 52] to object detection [55, 56] and segmentation [22, 33]. To ensure broad applicability in downstream tasks, visual encoders are typically pretrained on generic objectives such as supervised image classification [21, 67], self-supervised learning [6, 8], or contrastive vision-language training [53, 76]. Their primary goal is to capture the semantic content of images, encoding information about objects, attributes, and other relevant features.

Given their central role in computer vision tasks, ensuring the robustness of visual encoders across diverse image distributions is crucial. This has led to extensive research on their sensitivity to various transformations and corruptions, including photometric and geometric changes [23], noise injection [23], natural corruptions [23], image compression [7, 17], adversarial perturbations [18, 36] and examples [25], and out-of-distribution inputs such as stylized images [16] or samples from unseen datasets [25].

This work investigates robustness in the era of foundation models, which are typically kept frozen after large-

[*]Equal contribution.

scale pretraining to preserve their acquired knowledge. However, this practice also preserves any inherent sensitivities or irregularities in their representation spaces. Our study differs from prior work in several key aspects:

1. We examine not only image-processing variations but also acquisition-related factors. We categorize these into two types: **image processing parameters**, which stem from standard preprocessing and encoding techniques, and **image acquisition parameters**, which originate from camera settings at capturing time. Therefore, images carry not only their semantic labels but also *metadata labels* that reflect the values of these acquisition and processing parameters.
2. While existing robustness studies primarily focus on severe corruptions and transformations that degrade the semantic signal, we investigate subtle variations that are often imperceptible to the human eye.
3. Prior work typically assumes a clean image state during training and evaluates robustness under test-time distribution shifts involving uncommon or degraded states. In contrast, we consider a continuum of valid states, characterized by metadata labels, emphasizing the interplay between the prior distribution of semantic and metadata labels, a key aspect of our findings.

To clarify these issues, we address the following questions: *How do metadata labels influence the representations generated by visual encoders?* and *Do metadata labels affect the quality of downstream task predictions?*

Our analysis spans a diverse range of visual encoders, covering different architectures, model sizes, training objectives, and datasets. Key findings include:

1. Acquisition and processing parameters leave identifiable traces in many models' representation spaces. Such traces allow the inversion, *i.e.* prediction, of the original input image's parameters with considerably higher accuracy than random chance.
2. These traces can overshadow semantic content in downstream tasks, leading to biases where model predictions are influenced by the acquisition or processing parameters rather than the actual object semantics (see Fig. 1). Interestingly, we find that similarity in the representation space depends on both semantic and metadata labels (see Fig. 2), with the latter often having a disruptive influence in downstream tasks.
3. We identify contrastive vision-language models as the most sensitive to such influences, and our experiments suggest that this sensitivity stems from the absence of strong image augmentations during pretraining.

These findings raise concerns about model reliability, especially in corner cases that, while artificially injected to our experiments, may occur in real-world applications.

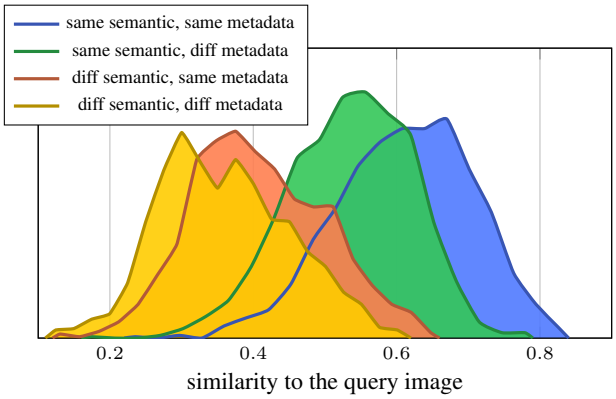

Figure 2. **Distribution of similarities with respect to a query image.** Four distributions are shown, based on whether the semantic and metadata labels of the images match those of the query. Metadata labels are based on JPEG quality. The results highlight that similarity is influenced by both types of labels.

2. Preliminaries

Visual encoders We analyze 47 visual encoders, each used to extract image embeddings without any further finetuning. We categorize them into three groups according to their training schemes:

- **Supervised** (SUP) are models trained with supervision for image classification on ImageNet [57]. Several variants of convolutional neural networks (CNN) and Transformer architectures are considered, including ResNet [21], ConvNeXt [42], and ViT [12].
- **Self-supervised learning** (SSL) are models trained with contrastive learning. Several variants of DINO [6], DINOv2 [50], and MoCov3 [9] are considered.
- **Contrastive visual-language** (CVL) are models trained for vision-language alignment. We use several variants of CLIP [53], OpenCLIP [10], and SigLIP [71, 76].

The full list of visual encoders and their settings can be found in the supplementary material.

Image processing parameters are the properties of an image that are not determined at the time of capture and that can be changed in post-processing stages. These include image compression (*e.g.* JPEG quality), color-based transformations (*e.g.* sharpening), image resizing (*e.g.* type of interpolation), *etc*. We use the notation $\mathcal{P} = \{p_1, \cdots, p_M\}$ to indicate the M possible classes for one of four processing parameters: JPEG compression, sharpening, amount of resizing, and resize interpolation.

Image acquisition parameters are the properties of an image that are determined by the method of its capture. These include specification of the camera (*e.g.* camera

Table 1. Image processing (proc.) and image acquisition (acq.) parameters under analysis and their number of classes.

parameter	type	class	description
JPEG compression	proc.	6	amount of JPEG compression
sharpening	proc.	3	amount of sharpening
resizing	proc.	3	amount of resizing
interpolation	proc.	4	type of interpolation during resize
make	acq.	9	manufacturer of the camera
model (all)	acq.	88	specific camera model used
model (smart)	acq.	12	specific smartphone used
model (smart vs non-smart)	acq.	2	whether camera is a smartphone
exposure	acq.	16	amount of light captured by sensor
aperture	acq.	17	size of the opening in the lens
ISO speed	acq.	16	camera sensor's sensitivity to light
focal length	acq.	13	distance from lens to sensor

maker or model) and camera settings used during capturing (*e.g.* exposure, aperture). We use the notation $\mathcal{A} = \{a_1, \dots a_M\}$ to indicate the M possible classes for one of eight acquisition parameters: make, model (all), model (smart), model (smart vs non-smart), exposure, aperture, ISO speed, and focal length.

Tab. 1 shows the list of parameters under analysis. More details are provided in the supplementary material.

Datasets Image processing parameters can be controlled and adjusted as desired, allowing us to study them on any existing dataset.[1] We use two common image classification datasets that carry semantic labels: ImageNet [57] and iNaturalist 2018 [27, 28], following the default splits into training ($\mathcal{X}_{\text{train}}$) and test ($\mathcal{X}_{\text{test}}$) sets. In contrast, image acquisition parameters are fixed once an image is captured. They can typically be recovered from the image's Exif tags, which most datasets do not include. We collect and make available two new datasets: FlickrExif and PairCams.

- **FlickrExif**: We use the Flickr API[2] to download images and their accompanying Exif metadata. For each month from January 2000 to August 2024, we collect between $2,000$ and $4,000$ safe-for-work photos with permissive licenses. To prevent the dataset from being dominated by a small group of prolific photographers, we limit the number of images contributed by each user to 10 per month and year. This leads to a total of $356,459$ images.
- **PairCams**: To ensure complete control over the data and eliminate potential semantic confounders, we manually collect a dataset with 730 pairs of photos, totaling $1,460$ images. Each pair depicts the same object or scene captured under identical conditions but different *camera types*. Specifically, we use two distinct categories of cameras: (i) modern smartphones, and (ii) older digital cameras.[3] All images are taken from the same angle, without flash, with automatic white balancing, automatic exposure mode, and Program AE as the exposure program. Fig. 3 shows examples of the collected pairs. The detailed list of cameras used for the collection is provided in the supplementary material.

[1] Images in existing datasets are already processed, so our processing is applied on top of any prior one.

[2] https://www.flickr.com/services/api/

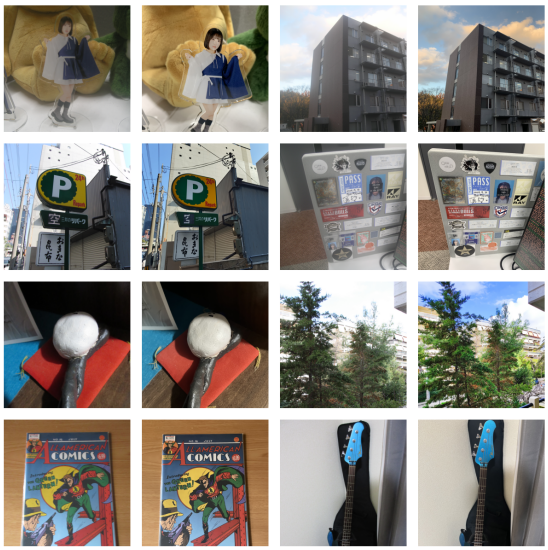

Figure 3. **Examples of images in the PairCams dataset.** Each pair depicts the same object and/or scene but taken with two different camera types. For each pair, the left image corresponds to a non-smartphone, and the right one to a smartphone.

3. Processing and acquisition traces

We examine whether visual encoders retain traces of image processing and acquisition parameters. A classifier is trained to predict the metadata labels. If its performance is close to random accuracy, it indicates that the embeddings do not encode information related to processing or acquisition parameters. If the classifier performs better than random chance, it shows that information about these parameters is indeed encoded within the embeddings.

3.1. Prediction of processing-based labels

Setup Given a training set $\mathcal{X}_{\text{train}}$ and a test set $\mathcal{X}_{\text{test}}$, we process every training image $x \in \mathcal{X}_{\text{train}}$ with $p_x \sim \text{Uniform}(\mathcal{P})$, where $\text{Uniform}(\mathcal{P})$ is the uniform distribution over $\mathcal{P}$, while we process all test images using $p_i \in \mathcal{P}$. More specifically, we make sets $\mathcal{D}_{\text{train}} = \{(x, p_x) | x \in \mathcal{X}_{\text{train}}, p_x \sim \text{Uniform}(\mathcal{P})\}$ and $\mathcal{D}_{\text{test}}(p_i) = \{(x, p_i) | x \in \mathcal{X}_{\text{test}}\}$ for a given p_i. We then train a linear classifier to predict the metadata label. We use 20% of the training set as the validation set to select the best learning rate and weight de-

[3] All smartphones in PairCams were released in 2018 or later. All digital cameras were released in 2014 or earlier.

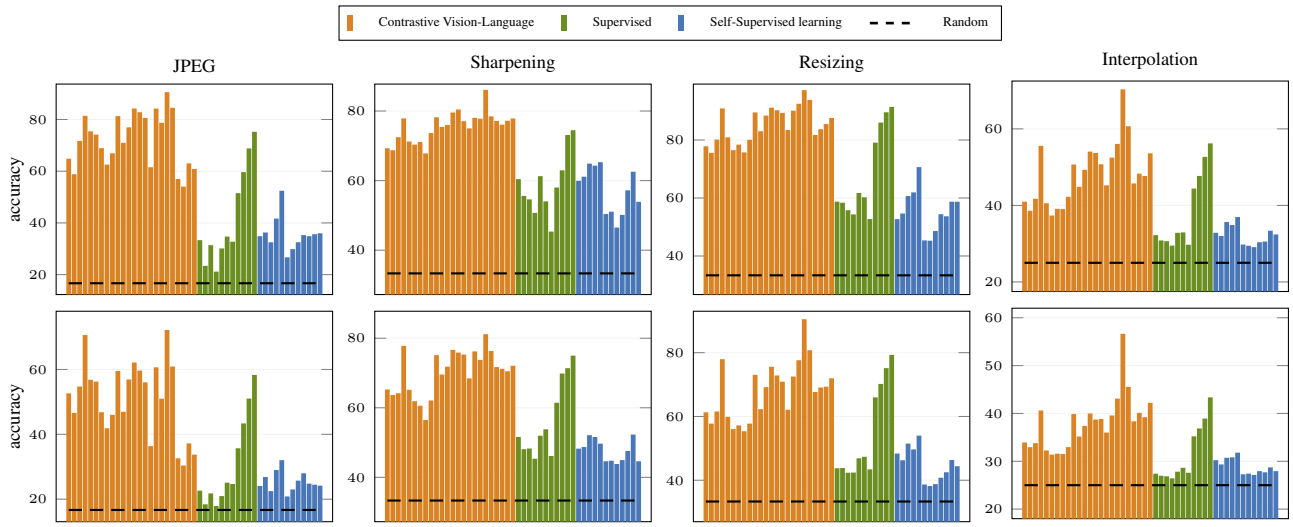

Figure 4. **Image processing-based label prediction.** Classification accuracy using a linear classifier on embeddings of different frozen visual encoders on ImageNet (top) and iNaturalist (bottom) datasets. Ordering is according to Tab. C in the supplementary material.

cay over 30 trials following [59].[4] After finding the best hyperparameters, we re-train the classifier on the whole training set. We report the classification accuracy on the test set averaged over every possible choice of p_i, as well as across 10 different seeds that control the sampling of p_x.[5]

Results Fig. 4 shows the linear classification accuracy for predicting processing-based labels. We observe that CVL models, compared to other types of models, show a higher ability to recognize all processing-based labels; *e.g.* CVL models are capable of exceeding 80% accuracy for predicting JPEG compression, sharpening, and resizing on ImageNet. Specifically, one version of a CVL model based on ConvNeXt performs the best for all processing types. Additionally, many supervised models, mostly based on ConvNeXt, also obtain a high performance. On the other hand, self-supervised models generally perform the worst on all processing parameters. Regarding different processing types, predicting interpolation is the hardest across all model types. However, the best performing CVL and supervised models still perform well above the random accuracy of 25% on both datasets.

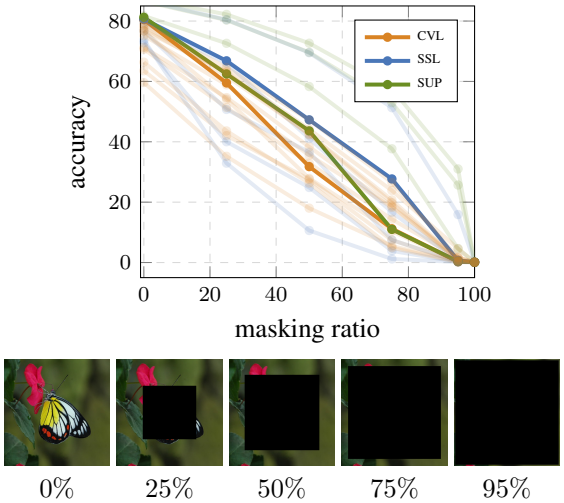

Figure 5. **ImageNet validation accuracy at different masking ratios.** 95% is enough to remove the ability for successful semantic label predictions, reducing the leakage of semantic cues into the acquisition-related label prediction task. At the bottom are visual examples of different masking ratios.

3.2. Prediction of acquisition-based labels

Setup For each acquisition parameter $\mathcal{A}$, we create a training set $\mathcal{X}_{\text{train}}$ and a test set $\mathcal{X}_{\text{test}}$. Except for model (all) and model (smart), we only use values of $\mathcal{A}$ with at least $5,000$ images. From these, we randomly select 500 images for the test set. The remaining images are undersampled to the size of the minority class of $\mathcal{A}$, with 200 images per class also used for validation. For the model (all) and model (smart) parameters, due to the limited amount of data per class, we instead select classes with at least 500 images, randomly select 500 of these images per class, and divide these into $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{text}}$ at a $4:1$ ratio, with 20% of $\mathcal{X}_{\text{train}}$ used for validation. We ensure that photographers are disjoint between the splits. We then use a linear classifier to predict the value of $\mathcal{A}$ from precomputed visual embeddings. Similarly to Sec. 3.1, we select the best hyperparameters on the validation set and re-train the classifier on the whole training set. To avoid spurious correlations between acquisition-

[4] Our implementation is based on https://github.com/naver/trex/tree/master/transfer

[5] Hyperparameters are tuned on the sampling of p_x of the first seed.

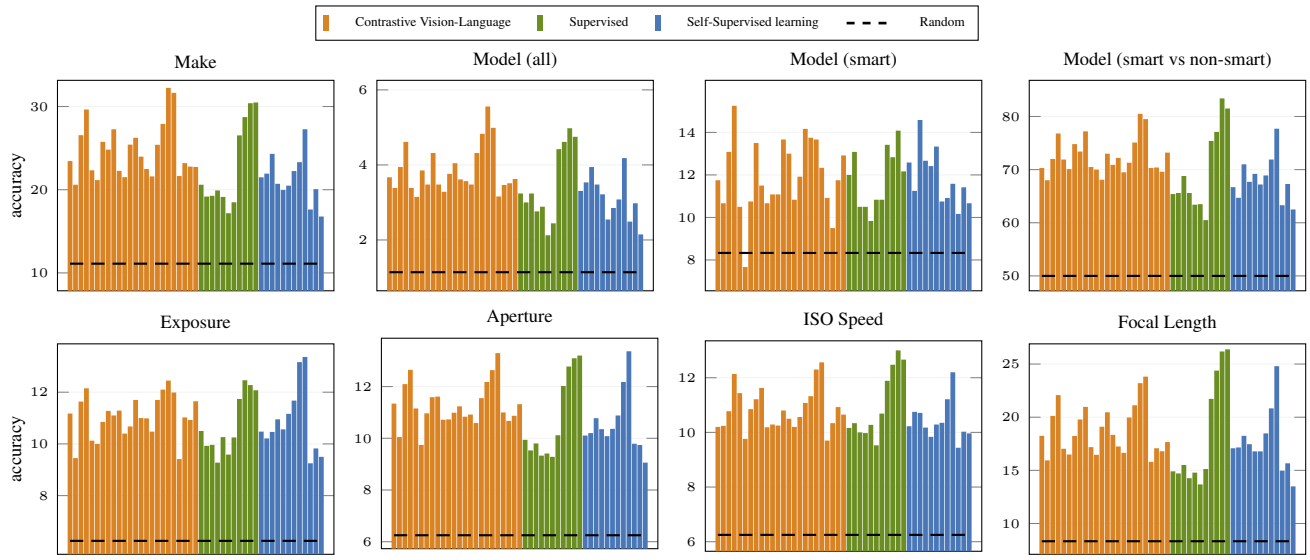

Figure 6. **Image acquisition-based label prediction.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders with images masked at 95% on FlickrExif dataset. Ordering is according to Tab. C in the supplementary material.

Table 2. Percentage of top-k neighbors with the same model (smart vs non-smart) as the query, averaged across all queries. Results on FlickrExif.

model	variant	class	$k = 10$	$k = 25$	$k = 50$
CLIP	ViT-L/14@336	CVL	70.0	69.1	67.6
SigLIP	ViT-B/16	CVL	59.0	57.5	56.4
ConvNeXt	ConvNeXt-B	SUP	57.1	55.8	55.2
ViT	ViT-H/14	SUP	53.1	53.0	52.4
MoCov3	ViT-B	SSL	58.3	56.3	54.8
DINOv2	ViT-L/14	SSL	54.1	53.7	53.0

based labels and semantic-based labels (*e.g.*, photos of birds taken with a specific focal length), we deliberately suppress semantic information by center-masking 95% of the image content.[6] Fig. 5 shows the ImageNet validation accuracy for multiple visual encoders when images are center-masked at ratios from 0 to 100%.

Results Fig. 6 illustrates prediction accuracies for acquisition-based labels on the FlickrExif test sets. Even with 95% masking, visual encoders predict acquisition parameters well above random chance. Results are generally consistent with Sec. 3.1 given that the highest accuracies tend to belong to CVLs and supervised ConvNeXts, albeit the best performance on some attributes belong to SSLs, and the gaps between training schemes are not as wide. All CVLs exceed 20% accuracy for make against a baseline of 11.11%, with many exceeding 25% and some passing 30%.

[6]In contrast to the version of this paper that appeared at ICCV 2025, whenever masking is applied we set all encoders' resize sizes and center crop sizes to be equal, and mask after cropping. Furthermore we now use 95% masking, equivalent to 90% masking in the ICCV version.

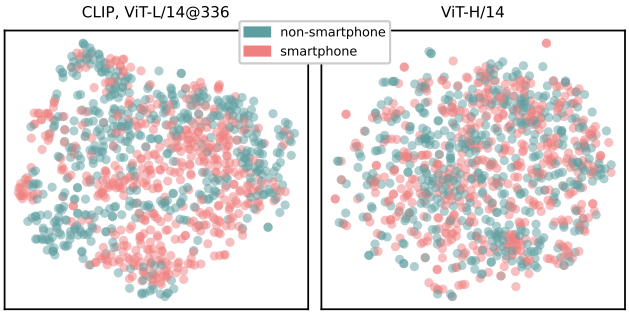

Figure 7. **t-SNE visualizations.** for two different visual encoders. Colors identify images by model (smart vs non-smart).

In contrast, not all SUP models can reach 20%, and no SSL model passes 30%. Furthermore, Tab. 2 shows the rate at which an image's top-k neighbors share its acquisition-based label for model (smart vs non-smart) across different encoders, without masking. The highest rates are from CVLs, which range from 59.0% to 70.0% compared to at most 58.3% from non-CVLs. The implication is that while visual encoders are storing model information in their embeddings, this trend is strongest among CVLs. This is visualized in the t-SNE [72] representations in Fig. 7, which shows how the most and the least discriminative encoders in Tab. 2 distribute images in their embedding spaces. While CLIP, a CVL model, shows areas of high concentration of images of similar model (smart vs non-smart), ViT, a supervised model, shows a much more even mix of points. This trend is further supported in Sec. 4.

4. Effect of traces on downstream tasks

While we have shown that metadata labels can be predicted from image embeddings, in this section, we inves-

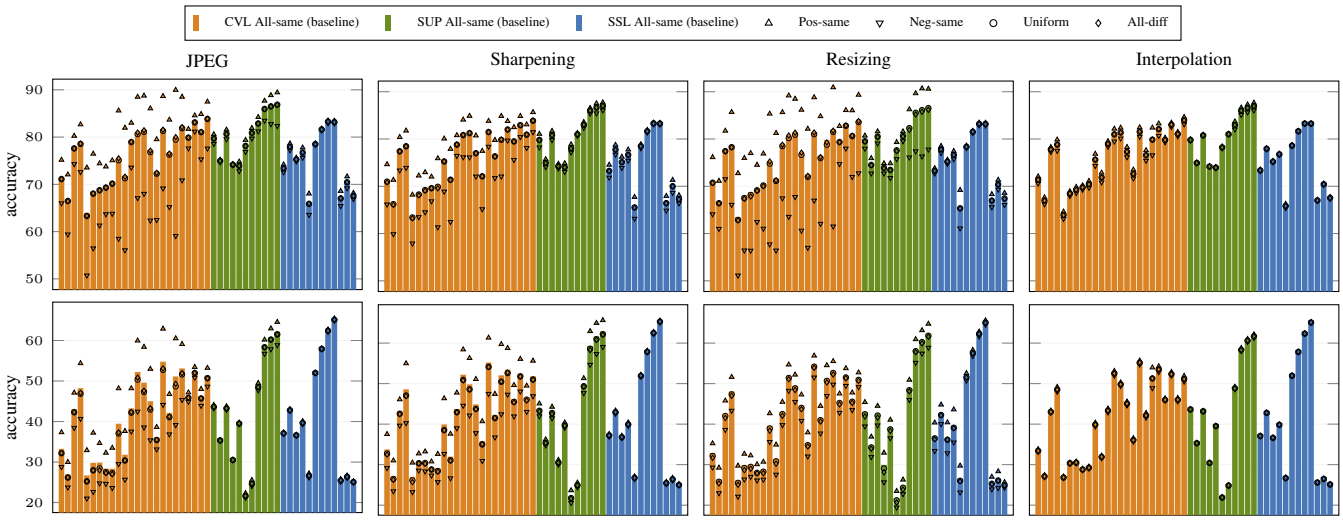

Figure 8. **Impact of image processing parameters on semantics.** Semantic label prediction accuracy on ImageNet (top) and iNaturalist (bottom) datasets in five different setups. *All-same (baseline)*: all training and test images share the same processing-based metadata label. *All-diff*: training images have the same metadata label, which is different than that of the test image. *Pos-same*: training images that are semantically positive to the test image have the same metadata label as the test image. *Neg-same*: training images that are semantically negative to the test image have the same metadata label as the test image. *Uniform*: the metadata labels are uniformly assigned to the training images. *Pos-same* and *neg-same* settings require an artificially created experiment where a different training set is used per test image. Shown for $k = 10$ and $k = 1$ for the kNN classifier for ImageNet and iNaturalist, respectively.

tigate whether this information can impact downstream applications that rely on the ability to predict semantic labels. We focus on two main tasks: image classification and near-duplicate image retrieval.

4.1. Processing distracts semantic predictions

To analyze the impact of processing parameters on semantic predictions, we use a kNN classifier to predict the semantic label of each image. In that way, we assess the representation capability of the models to bring closer images with the same semantic label.

Setup Given a training set $\mathcal{X}_{\text{train}}$ and a test set $\mathcal{X}_{\text{test}}$, the same as in Sec. 3.1, we measure the classification accuracy of predicting the semantic label. We perform this in five setups, where *a separate training set is formed per test image*.

(i) *All-same (baseline):* processing all training $\mathcal{X}_{\text{train}}$ and test $\mathcal{X}_{\text{test}}$ images with an identical processing value p_i. It serves as a baseline for reference, where classification performance is fully determined by the semantic similarity between image embeddings.

(ii) *All-diff:* processing all training $\mathcal{X}_{\text{train}}$ images with processing value p_j and test $\mathcal{X}_{\text{test}}$ images with processing value p_i, for $i \neq j$. It reflects the common setup of existing robustness studies [23].

(iii) *Pos-same:* processing a test image $x \in \mathcal{X}_{\text{test}}$ and training images that share the same semantic label (positive) as x with p_i, while processing training images that have a different semantic label (negative) with processing value p_j with $i \neq j$. It investigates whether semantic positives (negatives) being metadata positives (negatives) affects the accuracy, compared to the baseline.

(iv) *Neg-same:* processing a test image $x \in \mathcal{X}_{\text{test}}$ and training images that do not have the same semantic label (negative) as x with p_i, while processing training images that have the same semantic label (positive) as x with processing value p_j with $i \neq j$. It investigates whether semantic positives (negatives) being metadata negatives (positives) affects the accuracy, compared to the baseline.

(v) *Uniform:* similarly to Sec. 3.1, processing $x \in \mathcal{X}_{\text{test}}$ with p_i while $x' \in \mathcal{X}_{\text{train}}$ is processed with $p_{x'} \sim \text{Uniform}(\mathcal{P})$. This reflects a more realistic scenario.

We report classification accuracy averaged across all possible combinations of p_i and p_j for each setup, as well as for 10 random seeds for uniform sampling of $p_{x'}$.

Results Fig. 8 illustrates the impact of metadata labels related to processing parameters on semantic recognition. When semantic and metadata labels are coupled and change together (*pos-same*, *neg-same*), performance is significantly affected for certain processing types and models. Although these setups are extreme, they represent corner cases that could still arise in real-world scenarios for a specific test image. JPEG compression has the greatest impact, followed by sharpening and resizing. Interpolation mostly has an insignificant effect on semantics. The greatest sensitivity is observed in CVLs, followed by some supervised models, mainly the ones based on the

Figure 9. **Visualization of top-10 nearest neighbors of a query.** in three different setups: *All-same (baseline)* (top), *Pos-same* (middle), *Neg-same* (bottom) using CVL ConvNeXt-L model. Green/red borders indicate a positive/negative image to the query according to the semantic label, while the icons on the left indicate the metadata label (JPEG compression) for the **query**, **positive**, and **negative** images.

ConvNeXt architecture. Self-supervised models are the least affected, providing hints on heavy geometric and photometric training augmentations causing the different behavior. The *all-diff* and *uniform* setups perform similarly, typically slightly below the baseline. This suggests that the perspective presented in this study differs from the traditional robustness assessments in prior research.

We visualize the top-10 nearest neighbors of a query image in three different setups in Fig. 9 for one of the most impacted models, *i.e.* the CVL ConvNeXt-L model. The traces of metadata labels in the representation space significantly affect the quality of the nearest images. Evidently, distance in the representation space is a function of both semantic and metadata labels for some foundation models.

4.2. Camera model distracts semantic retrieval

Using the PairsCams dataset, we investigate the impact of image acquisition parameters, in particular whether the camera was a modern smartphone, on the task of near-duplicate image retrieval.

Setup Given a query image of an object or scene captured by one camera type, the objective is to retrieve the corresponding image of the same object or scene (positive) taken by a different camera type. We consider two different settings for the negative images in the retrieval database: (i) *same* – negatives are captured by the same camera type as the query, or (ii) *different* – negatives are captured by a different camera type. We evaluate performance based on the average recall@1 under both settings.

Results Results are presented in Fig. 10. All points are above the diagonal, suggesting the influence of the metadata label on the retrieval performance. Vision encoders consistently fail to retrieve the image of the same object or scene in the presence of negatives taken with the same camera type. The experiment reveals that CVLs have the highest disparity between the two settings. A model with near-perfect recall@1 can see a performance drop below

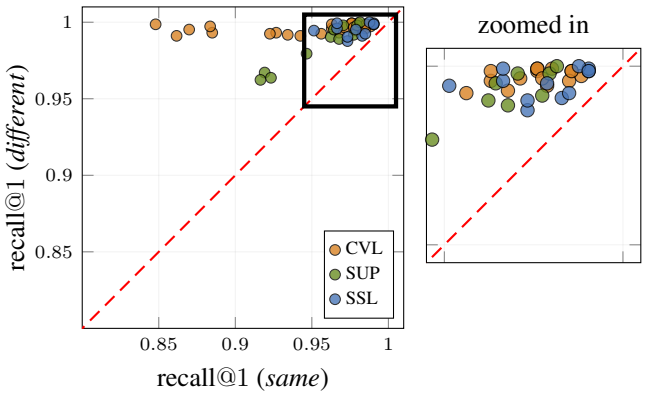

Figure 10. **Impact of image acquisition parameters on semantics.** Retrieval performance measured by recall@1 using the PairsCams dataset for the settings where the negatives are captured either by the *same* or a *different* camera type than the query. Each point corresponds to a different visual encoder. The diagonal line is provided as a reference to indicate a lack of influence from the metadata label, *i.e.* model (smart vs non-smart).

0.85 simply by having the negatives share the same camera type as the queries. See Fig. 1 for a qualitative example; the near-identical positive is not retrieved first due to the metadata label mismatch.

5. Related work

Robustness benchmarks and mitigation There has been much effort on the improvement of visual encoders' robustness under shifts from their training distributions. This includes the design of benchmarks that evaluate robustness on hard conditions [1, 16, 23–25, 39], with some specializing on corruptions and perturbations [23], stylization techniques [16], JPEG compressions [17], adversarial and out-of-distribution examples [25], attribute editing [39], and environment and sensor shifts [1]. Other work focuses specifically on CVL models under similar conditions *i.e.* adver-

sarial examples [46], perturbations [62], JPEG compression [7], and multiple aspects of robustness simultaneously [40]. Additionally, several solutions for improving robustness, involving the use of strong augmentations [16, 23], a more elaborate training process [79], learning artifact correction [14], test-time adaptation [64] and training [51, 74], or prompting [38] and adversarial fine-tuning [63], have been shown to improve CVL zero-shot performance. Unlike prior work, we analyze images' processing and acquisition attributes and the ability of pretrained models to capture such information. We show that such signals can dominate representations and distract semantics, even though they are imperceptible to the naked eye. Also, we do not consider training images derived from a "clean" in-domain distribution since all images carry their unique processing and acquisition attributes, with only a small fraction being controllable (*e.g.*, JPEG compression).

Analysis of visual encoder robustness Analyzing the behavior of visual encoders, Goodfellow *et al*. [18] investigate the impact of inserting human-imperceptible noise into test images during evaluation. It is found that by altering just one pixel, model predictions can be changed [69]. Other analysis involve texture-shape bias [16], translation/shift invariance and positional encoding [29, 30, 77], comparison between architecture families [48], or emergent properties from large-scale pretraining [66]. Closely related to our work is model performance assessment under different JPEG compression [7, 14], while recent work [19] highlights that discrepancies in JPEG compression can affect performance evaluation, resulting in misleading outcomes.

Attribute detection Similar to our experiments on metadata label prediction, previous work has explored attribute prediction tasks, such as camera model identification [4, 45, 54] or camera parameters prediction (*e.g*. focal length and radial distortion [73] or camera tilt and roll [43]). There is also work on aligning Exif tags with visual content via CVLs [78], as well as predicting image compression [13] and coding [37]. While our experiments share clear similarities, this typically involves training models with supervised labels to predict image attributes. In contrast, we reveal that this type of information is already contained in the visual representations, which can predict the processing and acquisition attributes without any further training or fine-tuning.

Visual bias Prior work [41, 70] has shown that dataset-specific biases allow a classifier to predict an image's dataset of origin well above random. To combat such kinds of biases, several methods for debiasing at training time have been proposed, like leveraging labels of attributes introducing bias [3, 26, 31, 58, 75] or pseudo-labels derived from the biased classifiers [2, 5, 47, 60]. More recent approaches employ external models to discover and then suppress biases by extracting common textual keywords from misclassified examples [32], or directly using large language models for bias discovery [11, 20, 61]. Our evaluation setup for analyzing the impact of metadata labels on semantic prediction shares similarities with approaches in this field [2, 26, 58].

6. Discussion and conclusion

Conclusions We have shown that metadata labels, such as acquisition and processing parameters, are encoded in the representations of foundational visual encoders, especially those trained with CVL loss. Our analysis revealed that traces of such labels can impact semantic recognition performance, depending on the relative distribution between semantic and metadata labels. This effect, however, is not uniform across all models: it is most pronounced in CVLs, while SSL encoders exhibit it to a lesser extent. This discrepancy may be attributed to the drastic pixel-wise and geometric augmentations in SSL, which CVLs do not employ. To verify this assumption, in Sec. A, we train from scratch and finetune a CVL model with and without augmentations. We show that the introduction of augmentations reduces the metadata information encoded in the representations.

Limitations While we have identified that metadata labels are encoded in foundational visual encoders and provided hints about potential causes, we cannot definitively pinpoint the source of the problem. Investigating this further is challenging due to the cost of retraining such models and the frequent use of private datasets and undisclosed implementation details. Although we do not propose specific mitigation techniques, we highlight the issue as an important area for future research.

Negative implications Metadata labels leaving traces in visual encoders to the point of overshadowing semantic information can lead to unpredictable outcomes, compromising generalizability, robustness, and potentially undermining the trustworthiness of the models. More critically, this effect could be exploited maliciously; for instance, an adversarial attack may manipulate metadata to intentionally mislead or deceive a model, posing risks in sensitive domains like healthcare, surveillance, or autonomous systems.

Positive implications There are some applications that can benefit from the implications revealed in this work. For example, recent methods for synthetic (deepfake) image detection leverage frozen CVL models for detection [49, 65] and localization [68]. Interestingly, CVL models can be used as a kNN classifier, *i.e*. without training a linear classifier [49]. We presume that the capability of CVL models to detect deepfakes shares the same underlying root enabling attribute-based detection.

Acknowledgments

This work was supported by the Junior Star GACR GM 21-28830M, the Horizon MSCA-PF grant No. 101154126, the Czech Technical University in Prague grant No. SGS23/173/OHK3/3T/13, JSPS KAKENHI No. JP23H00497 and JP22K12091, JST CREST Grant No. JPMJCR20D3, and JST FOREST Grant No. JPMJFR216O. We acknowledge VSB – Technical University of Ostrava, IT4Innovations National Supercomputing Center, Czech Republic, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (grant ID: 90254).

We thank Jan Franců, Yannis Kalantidis, Symeon Papadopoulos, and Vladimir Risojević for many helpful comments.

References

[1] Eunsu Baek, Keondo Park, Jiyoon Kim, and Hyung-Sin Kim. Unexplored faces of robustness and out-of-distribution: Covariate shifts in environment and sensor domains. In *CVPR*, 2024. 7, 3

[2] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *ICML*, 2020. 8

[3] Carlo Alberto Barbano, Benoit Dufumier, Enzo Tartaglione, Marco Grangetto, and Pietro Gori. Unbiased supervised contrastive learning. In *ICLR*, 2023. 8

[4] Belhassen Bayar and Matthew C Stamm. Towards open set camera model identification using a deep learning framework. In *ICASSP*, 2018. 8

[5] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, 2019. 8

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2

[7] Cangxiong Chen, Vinay Namboodiri, and Julian Padget. Understanding the vulnerability of clip to image compression. In *NeurIPSW*, 2023. 1, 8

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 2

[10] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 2, 1

[11] Moreno D'Incà, Elia Peruzzo, Massimiliano Mancini, Dejia Xu, Vidit Goel, Xingqian Xu, Zhangyang Wang, Humphrey Shi, and Nicu Sebe. Openbias: Open-set bias detection in text-to-image generative models. In *CVPR*, 2024. 8

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2

[13] Thierry Dumas, Aline Roumy, and Christine Guillemot. Context-adaptive neural network-based prediction for image compression. *TIP*, 2019. 8

[14] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Analyzing and mitigating jpeg compression defects in deep learning. In *CVPR*, 2021. 8

[15] Flickr. Real-time resizing of flickr images using gpus. https://code.flickr.net/2015/06/25/real-time-resizing-of-flickr-images-using-gpus/, 2015. 1

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2018. 1, 7, 8

[17] Sanjukta Ghosh, Rohan Shet, Peter Amon, Andreas Hutter, and André Kaup. Robustness of deep convolutional neural networks for image degradations. In *ICASSP*, 2018. 1, 7

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 8

[19] Patrick Grommelt, Louis Weiss, Franz-Josef Pfreundt, and Janis Keuper. Fake or jpeg? revealing common biases in generated image detection datasets. In *ECCVW*, 2024. 8

[20] Quentin Guimard, Moreno D'Incà, Massimiliano Mancini, and Elisa Ricci. Classifier-to-bias: Toward unsupervised automatic bias detection for visual classifiers. In *CVPR*, 2025. 8

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 1, 6, 7, 8

[24] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *CVPR*, 2021.

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021. 1, 7

[26] Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. In *NeurIPS*, 2021. 8

[27] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 3

[28] iNaturalist 2018 competition dataset. iNaturalist 2018 competition dataset. https://github.com/visipedia/inat_comp/tree/master/2018, 2018. 3

[29] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? In *ICLR*, 2020. 8

[30] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *CVPR*, 2020. 8

[31] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, 2019. 8

[32] Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. Discovering and mitigating visual biases through keyword explanation. In *CVPR*, 2024. 8

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1

[34] Giorgos Kordopatis-Zilos, Vladan Stojnić, Anna Manko, Pavel Suma, Nikolaos-Antonios Ypsilantis, Nikos Efthymiadis, Zakaria Laskar, Jiri Matas, Ondrej Chum, and Giorgos Tolias. ILIAS: Instance-level image retrieval at scale. In *CVPR*, 2025. 1

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[36] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLRW*, 2017. 1

[37] Jiahao Li, Bin Li, Jizheng Xu, Ruiqin Xiong, and Wen Gao. Fully connected network-based intra prediction for image coding. *TIP*, 2018. 8

[38] Lin Li, Haoyan Guan, Jianing Qiu, and Michael Spratling. One prompt word is enough to boost adversarial robustness for pre-trained vision-language models. In *CVPR*, 2024. 8

[39] Xiaodan Li, Yuefeng Chen, Yao Zhu, Shuhui Wang, Rong Zhang, and Hui Xue. Imagenet-e: Benchmarking neural network robustness via attribute editing. In *CVPR*, 2023. 7

[40] Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A Lee, Yuke Zhu, et al. Multibench: Multiscale benchmarks for multimodal representation learning. In *NeurIPS*, 2021. 8

[41] Zhuang Liu and Kaiming He. A decade's battle on dataset bias: Are we there yet? In *ICLR*, 2025. 8

[42] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2

[43] Manuel Lopez, Roger Mari, Pau Gargallo, Yubin Kuang, Javier Gonzalez-Jimenez, and Gloria Haro. Deep single image camera calibration with radial distortion. In *CVPR*, 2019. 8

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[45] Manisha, Chang-Tsun Li, Xufeng Lin, and Karunakar A Kotegar. Beyond prnu: Learning robust device-specific fingerprint for source camera identification. *Sensors*, 2022. 8

[46] Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *ICLR*, 2023. 8

[47] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *NeurIPS*, 2020. 8

[48] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *NeurIPS*, 2021. 8

[49] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, 2023. 8

[50] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 2, 1

[51] David Osowiechi, Mehrdad Noori, Gustavo Vargas Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail Ayed, and Christian Desrosiers. Watt: Weight average test time adaptation of clip. In *NeurIPS*, 2025. 8

[52] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *PAMI*, 2018. 1

[53] 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 *ICML*, 2021. 1, 2

[54] Abdul Muntakim Rafi, Thamidul Islam Tonmoy, Uday Kamal, QM Jonathan Wu, and Md Kamrul Hasan. Remnet: remnant convolutional neural network for camera model identification. *Neural Computing and Applications*, 2021. 8

[55] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[56] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1

[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2, 3

[58] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *ICLR*, 2020. 8

[59] Mert Bulent Sariyildiz, Yannis Kalantidis, Karteek Alahari, and Diane Larlus. No reason for no supervision: Improved generalization in supervised models. In *ICLR*, 2023. 4

[60] Ioannis Sarridis, Christos Koutlis, Symeon Papadopoulos, and Christos Diou. Flac: Fairness-aware representation learning by suppressing attribute-class associations. *PAMI*, 2024. 8

[61] Ioannis Sarridis, Christos Koutlis, Symeon Papadopoulos, and Christos Diou. MAVias: Mitigate any visual bias. In *ICCV*, 2025. 8

[62] Madeline Schiappa, Shruti Vyas, Hamid Palangi, Yogesh Rawat, and Vibhav Vineet. Robustness analysis of video-language models against visual and language perturbations. In *NeurIPS*, 2022. 8

[63] Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *ICML*, 2024. 8

[64] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020. 8

[65] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *SIGSAC*, 2023. 8

[66] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *ICCV*, 2023. 8

[67] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[68] Stefan Smeu, Elisabeta Oneata, and Dan Oneata. Declip: Decoding clip representations for deepfake localization. In *WACV*, 2025. 8

[69] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 8

[70] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 8

[71] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv*, 2025. 2

[72] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 5

[73] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Single-image Calibration with Geometric Optimization. In *ECCV*, 2024. 8

[74] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 8

[75] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *CVPR*, 2020. 8

[76] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 1, 2

[77] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 8

[78] Chenhao Zheng, Ayush Shrivastava, and Andrew Owens. Exif as language: Learning cross-modal associations between images and camera metadata. In *CVPR*, 2023. 8

[79] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *CVPR*, 2016. 8

Processing and acquisition traces in visual encoders: What does CLIP know about your camera?

Supplementary Material

A. Impact of augmentations on encoding of metadata in CVLs

To verify our assumption that one of the main reasons that SSL models generally encode less metadata information is the use of heavy augmentations, we train a CVL model using OpenCLIP [10] both without heavy augmentation (default OpenCLIP setup) and with DINOv2 [50] style color augmentations.[7] We train a ViT-B/32 CVL model both from scratch and finetuned from the LAION-2B checkpoint on the YFCC15M dataset [53]. We follow the default training hyperparameters from OpenCLIP, which is training for 32 epochs with a batch size of $32k$ using the AdamW [44] optimizer. We utilize the cosine scheduler with a warmup of $2,000$ iterations. Learning rate is set to $5e^{-4}$ and weight decay is equal to 0.2.

Fig. A and Fig. B present the results for processing-based and acquisition-based metadata label prediction, respectively. Augmentations greatly reduce the accuracy of processing-based metadata label prediction. We observe a similar trend for acquisition-based metadata labels, though not to the same extent and consistency.

Fig. C presents the results for the effects of processing on the downstream tasks. Augmentations greatly reduce the influence of processing parameters on the prediction of semantics.

Although these results point to the lack of heavy augmentations in CVLs as one of the main reasons for their strong encoding of metadata information, further investigation is still necessary. This is because some CVL models, like SigLIP [76], encode less metadata information on attributes like JPEG, although they do not employ heavy augmentations.

B. Parameters description

We consider the following processing and acquisition parameters.

- **JPEG compression** is one of the most common operations that will be applied to an image after its acquisition. JPEG applies lossy compression where the amount of compression can be controlled by the quality and chroma-subsampling parameters. To investigate the influence of JPEG compression on image representations, we recompress images using quality $\in \{75, 85, 95\}$ and chroma-subsampling $\in$ {4:2:0, 4:4:4}, which gives $|\mathcal{P}| = 6$ possible processing parameter values.
- **Sharpening** corrects pixel values such that the image appears sharper, and is commonly automatically applied by different services [15]. We use unsharp mask based sharpening of an image $\mathcal{I}$ given as $\text{sharp}(\mathcal{I}) = \alpha\mathcal{I} + (1-\alpha)\text{blur}(\mathcal{I})$ where α controls the sharpness of the image. $\alpha = 1$ gives the original image, while $\alpha > 1$ gives a sharper image. For $\mathcal{P}$ we consider a set of processing parameter values given when $\alpha \in \{1, 2, 4\}$.
- **Resizing** is a common operation applied to images, after their acquisition, that changes the dimension of the image. To evaluate the influence of resizing, we set processing parameter values as $\mathcal{P} = \{1\text{x}, 0.5\text{x}, 2\text{x}\}$ that define original image, image where both width and height are halved, and image where both width and height are doubled, respectively. We use bilinear interpolation.
- **Interpolation** defines the interpolation function used during image resizing. To evaluate the influence of interpolation function, we set processing parameter values as $\mathcal{P} = \{\text{bilinear}, \text{bicubic}, \text{lanczos}, \text{box}\}$, and we resize each image by changing its both sides by $r\%$ where $r \sim \text{Uniform}[-20, 20]$. [8]
- **Make** refers to the manufacturer of the camera, based on Exif metadata. Based on our setup, our analysis is based on nine manufacturers, namely *Apple*, *Canon*, *EASTMAN KODAK COMPANY*, *FUJIFILM*, *NIKON*, *OLYMPUS OPTICAL CO.,LTD*, *Panasonic*, *SONY*, and *Samsung*.
- **Model (all)** refers to the specific camera model used to capture the photo. We study 88 different camera models, shown in Tab. A.
- **Model (smart)** refers to the specific camera model used to capture the photo, but only among photos captured by smartphones. The 12 classes we study are also shown in Tab. A.
- **Model (smart vs non-smart)** is a binary parameter that indicates whether the camera used to shoot the photo was a smartphone. When analyzing this parameter, we use a subset of data that was curated to conveniently identify non-smartphones images and smartphone images. The former comprise all images taken with a camera manufactured by Canon, Nikon, Fujifilm, Panasonic, or Olympus; while the latter comprise all images taken with a smartphone manufactured by Apple, Google, Huawei, Xiaomi, or Motorola.
- **Exposure** refers to the amount of time that light was al-

[7] We use color jitter, random grayscaling, and random blurring.

[8] Note that the same value of r is applied per image across all different interpolations.

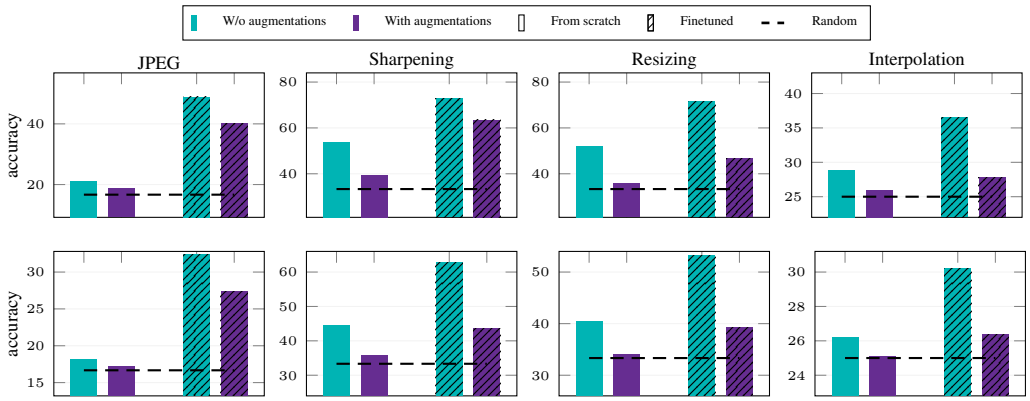

Figure A. **Image processing-based label prediction using a CVL model trained with and without augmentations.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders on ImageNet (top) and iNaturalist (bottom) datasets. CVL model trained both from scratch and finetuned starting from a LAION-2B checkpoint.

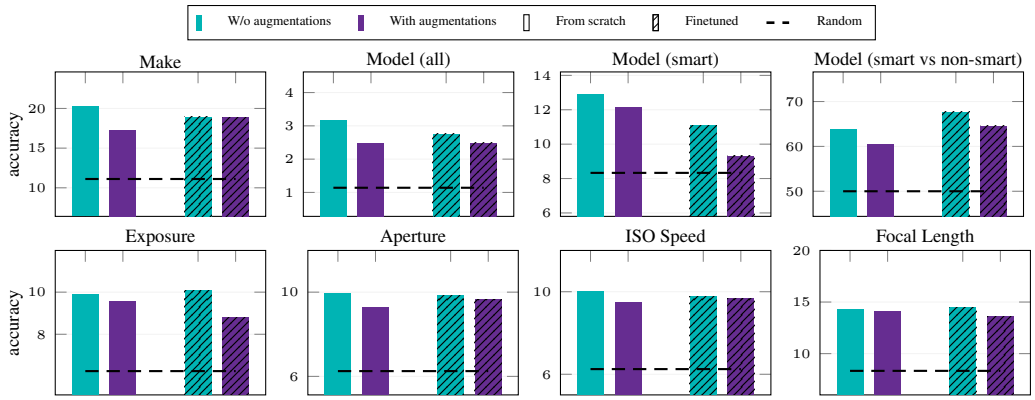

Figure B. **Image acquisition-based label prediction using a CVL model trained with and without augmentations.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders with images masked at 95% on the FlickrExif dataset. CVL model trained both from scratch and finetuned starting from a LAION-2B checkpoint.

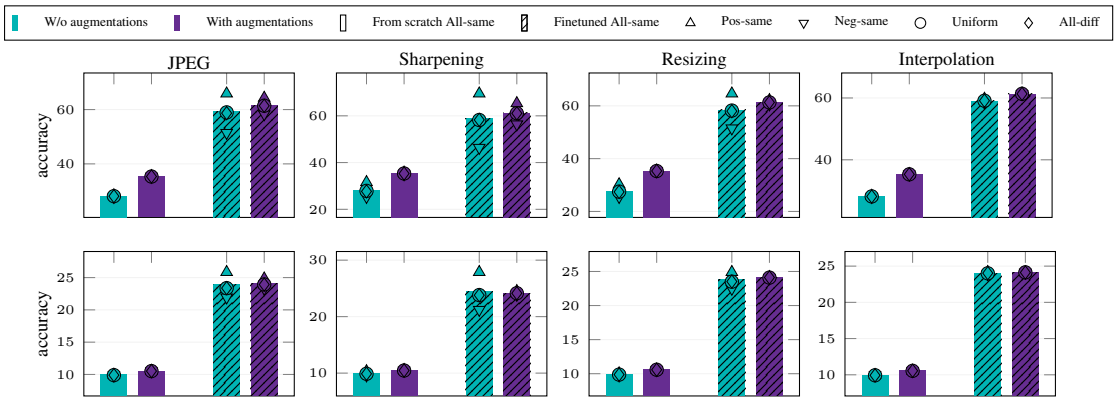

Figure C. **Impact of image processing parameters on semantics for a CVL model trained with and without augmentations.** Semantic label prediction accuracy on ImageNet (top) and iNaturalist (bottom) datasets in five different setups. *All-same (baseline)*: all training and test images share the same processing-based metadata label. *All-diff*: training images have the same metadata label, which is different than that of the test image. *Pos-same*: training images that are semantically positive to the test image have the same metadata label as the test image. *Neg-same*: training images that are semantically negative to the test image have the same metadata label as the test image. *Uniform*: the metadata labels are uniformly assigned to the training images. *Pos-same* and *neg-same* settings require an artificially created experiment where a different training set is used per test image. CVL model trained both from scratch and finetuned starting from a LAION-2B checkpoint. Shown for k = 10 and k = 1 for the kNN classifier for ImageNet and iNaturalist, respectively.

Table A. All 88 camera models studied when analyzing the acquisition attribute model (all). Names are presented as they were found in the Exif metadata. Underlined models refer to smartphones analyzed under the model (smart) parameter.

CYBERSHOT	Canon EOS DIGITAL REBEL	NIKON D3100	NIKON D800
Canon EOS 10D	Canon EOS DIGITAL REBEL XT	NIKON D3200	NIKON D810
Canon EOS 20D	Canon EOS R	NIKON D40	NIKON D850
Canon EOS 300D DIGITAL	Canon EOS R5	NIKON D5	NIKON D90
Canon EOS 30D	Canon EOS R6	NIKON D50	NIKON Z 6
Canon EOS 350D DIGITAL	Canon EOS REBEL T3i	NIKON D500	NIKON Z 6_2
Canon EOS 40D	Canon EOS Rebel T6	NIKON D5000	NIKON Z 9
Canon EOS 450D	Canon EOS-1D X	NIKON D5100	X-T2
Canon EOS 50D	Canon EOS-1D X Mark II	NIKON D5200	X-T3
Canon EOS 5D	E-M1MarkII	NIKON D5300	X-T4
Canon EOS 5D Mark II	E5700	NIKON D5500	iPhone 11
Canon EOS 5D Mark III	E990	NIKON D5600	iPhone 11 Pro Max
Canon EOS 5D Mark IV	ILCE-6000	NIKON D600	iPhone 12 Pro
Canon EOS 600D	ILCE-6400	NIKON D610	iPhone 12 Pro Max
Canon EOS 60D	ILCE-7	NIKON D70	iPhone 13 Pro
Canon EOS 6D	ILCE-7M3	NIKON D700	iPhone 6
Canon EOS 6D Mark II	ILCE-7RM2	NIKON D7000	iPhone 6s
Canon EOS 70D	ILCE-7RM3	NIKON D7100	iPhone 7
Canon EOS 7D	Kodak CLAS Digital Film Scanner / HR200	NIKON D7200	iPhone 7 Plus
Canon EOS 7D Mark II	NIKON D100	NIKON D750	iPhone X
Canon EOS 80D	NIKON D200	NIKON D7500	iPhone XR
Canon EOS 90D	NIKON D300	NIKON D80	iPhone XS

lowed to enter the camera while taking the photo. This is a rational number, which in our data ranges from $1/1,000$ seconds to $1/30$ seconds.

- **Aperture** refers to size of the opening in the lens and the corresponding amount of light thus allowed to enter the camera while taking the photo. This is measured in f-numbers, calculated as the ratio between the focal length and the diameter of the lens opening. In our experiments, these ratios range from 1.8 to 11.
- **ISO Speed** is a parameter that measures the camera sensor's sensitivity to light, with higher values leading to higher sensitivity and lower values leading to lower sensitivity. In our data, these numbers range from 50 to 3200.
- **Focal Length** describes the distance between the center of the camera's lens and the camera's sensor. This is typically measured in mm. Our data covers focal lengths ranging from 4 mm to 200 mm.

C. PairsCams dataset

Cameras used to collect the PairsCams dataset are shown in Table B with the number of images taken by each camera.

Table B. Cameras used during data collection. Each object or scene is captured by two cameras, leading to a total of 1,460 photos from 730 pairs.

model	type	year	images
iPhone XR	smartphone	2018	295
Canon IXY 630	compact	2014	285
Pixel 4	smartphone	2019	190
iPhone SE (3rd generation)	smartphone	2022	120
Sony Cyber-shot DSC-WX300	compact	2013	120
Olympus C-8080 Wide Zoom	compact	2004	100
Casio Exilim EX-FH20	compact	2008	100
Canon EOS 450D	DSLR	2008	80
Xiaomi Poco X5 Pro	smartphone	2023	40
iPhone 12	smartphone	2020	26
iPhone 14 Pro	smartphone	2022	25
Olympus μ700	compact	2006	25
Nothing Phone (2)	smartphone	2023	20
iPhone 12 Pro	smartphone	2020	14
Motorola Moto G XT1032	smartphone	2013	10
Nikon Coolpix S200	compact	2007	10
total			1,460

D. Visual encoders

Our visual encoders are acquired from the following repositories: OpenAI,[9] OpenCLIP,[10] timm,[11] and FAIR.[12] We follow each encoder's default preprocessing to extract image representations. This typically involves resizing images based on their smaller side, followed by center-cropping to the encoder's input resolution, and normalization of the image tensor.

[9] https://github.com/OPENAI

[10] https://github.com/mlfoundations/open_clip

[11] https://github.com/huggingface/pytorch-image-models

[12] https://github.com/facebookresearch/moco-v3

E. Additional results on ImageNet-ES

An existing dataset that can be used for acquisition label prediction is ImageNet-ES [1], which contains ImageNet images that have been recaptured under varying acquisition settings, including ISO, shutter speed, aperture, and lighting conditions. Originally designed for out-of-distribution detection, the dataset features disjoint test and training labels. Thus, we randomly split the provided validation set into our own training and test sets using a 9:1 ratio. We follow the hyperparameter tuning and training protocol described in

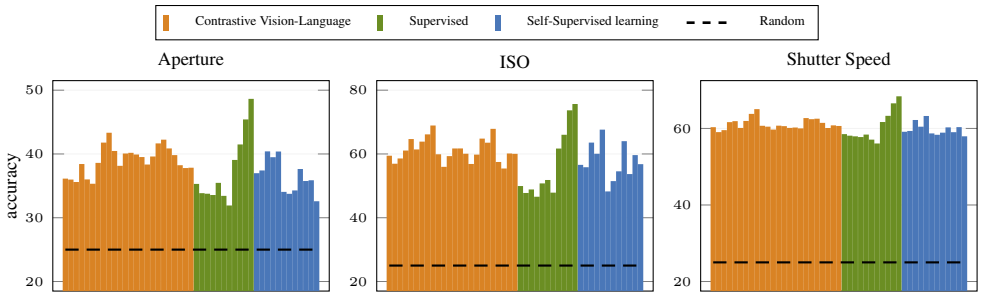

Figure D. **Image acquisition-based label prediction on ImageNet-ES.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders with no masking. Ordering is according to Tab. C.

Sec. 3, using 12.5% of the training set for validation. Each image is annotated with metadata labels according to their aperture, ISO, and shutter speed labels, formulating a four-class classification task for each attribute.

Fig. D presents the performance of classifiers trained on frozen embeddings for the prediction of each attribute. Models across all categories achieve classification accuracy well above random chance. We attribute this to the broad range of values of the acquisition parameters used to create ImageNet-ES. For example, the ISO values of ImageNet-ES's validation set ranges from 200 to $12,800$, while the corresponding values in FlickrExif only range from 50 to $3,200$. This wider range may result in more visually distinguishable cases compared to those in FlickrExif.

F. Effect of masking

In this section, we assess the impact of the masking applied for the prediction of acquisition labels in Sec. 3.2. Fig. E and Fig. F show the classification accuracy on FlickrExif with 0% and 75% masking, respectively. Comparing the two figures with Fig. 6, we observe that retaining semantic information in the input images makes it easier to identify acquisition labels, leading to higher classification accuracy. This suggests potential correlations between acquisition labels and semantic content, which can be exploited by the models to achieve a better performance.

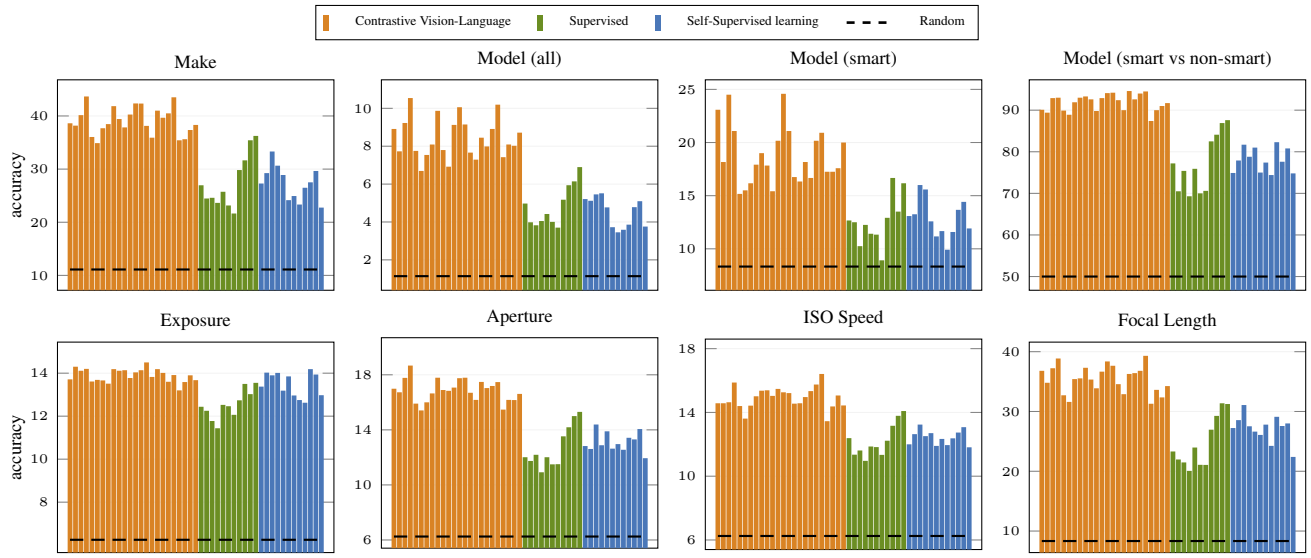

Figure E. **Image acquisition-based label prediction on FlickrExif without masking.** Classification accuracy using a linear classifier. Ordering is according to Tab. C.

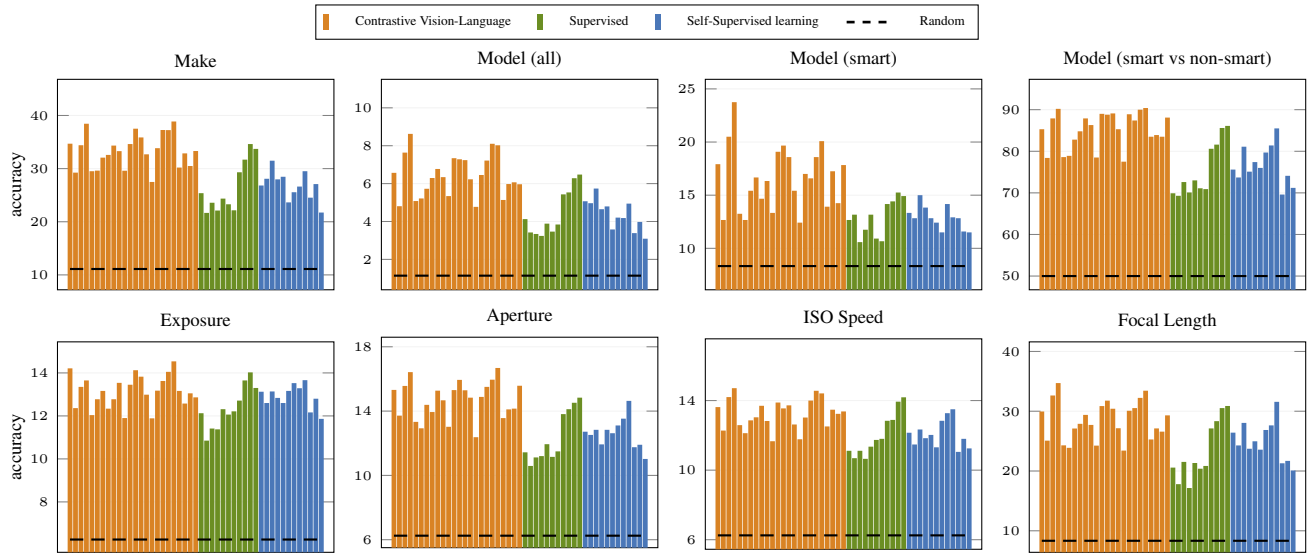

Figure F. **Image acquisition-based label prediction on FlickrExif with a** 75% **masking ratio.** Classification accuracy using a linear classifier. Ordering is according to Tab. C.

Table C. List of all visual encoders used with their characteristics.

id	model	variant	arch	class	dim	resolution	params (M)	train dataset
1	CLIP	ViT-B/16	Transformer	CVL	512	224	88	WIT
2	CLIP	ViT-B/32	Transformer	CVL	512	224	86	WIT
3	CLIP	ViT-L/14	Transformer	CVL	768	224	304	WIT
4	CLIP	ViT-L/14@336	Transformer	CVL	768	336	304	WIT
5	CLIP	RN50	CNN	CVL	1024	224	38	WIT
6	CLIP	RN101	CNN	CVL	512	224	56	WIT
7	CLIP	RN50×4	CNN	CVL	640	288	87	WIT
8	CLIP	RN50×16	CNN	CVL	768	384	167	WIT
9	CLIP	RN50×64	CNN	CVL	1024	448	420	WIT
10	OpenCLIP	ViT-B/16	Transformer	CVL	512	224	86	LAION-2B
11	OpenCLIP	ViT-B/32	Transformer	CVL	512	224	87	LAION-2B
12	OpenCLIP	ViT-L/14	Transformer	CVL	768	224	303	LAION-2B
13	OpenCLIP	ViT-H/14	Transformer	CVL	1024	224	632	LAION-2B
14	OpenCLIP	ViT-g/14	Transformer	CVL	1024	224	1012	LAION-2B
15	OpenCLIP	ViT-B/16	Transformer	CVL	512	224	86	DataComp-1B
16	OpenCLIP	ViT-B/32	Transformer	CVL	512	256	87	DataComp-1B
17	OpenCLIP	ViT-L/14	Transformer	CVL	768	224	303	DataComp-1B
18	OpenCLIP	ConvNeXt-B	CNN	CVL	640	256	88	LAION-2B
19	OpenCLIP	ConvNeXt-L	CNN	CVL	768	320	199	LAION-2B
20	OpenCLIP	ConvNeXt-XXL	CNN	CVL	1024	256	846	LAION-2B
21	SigLIP	ViT-B/16	Transformer	CVL	768	256	93	WebLI
22	SigLIP	ViT-L/16	Transformer	CVL	1024	256	316	WebLI
23	SigLIP2	ViT-B/16	Transformer	CVL	768	256	93	WebLI
24	SigLIP2	ViT-L/16	Transformer	CVL	1024	256	316	WebLI
25	ViT	ViT-B/16	Transformer	SUP	768	224	86	ImageNet-21k
26	ViT	ViT-B/32	Transformer	SUP	768	224	86	ImageNet-21k
27	ViT	ViT-L/16	Transformer	SUP	1024	224	307	ImageNet-21k
28	ViT	ViT-L/32	Transformer	SUP	1024	224	307	ImageNet-21k
29	ViT	ViT-H/14	Transformer	SUP	1280	224	632	ImageNet-21k
30	ResNet	RN50	CNN	SUP	2048	224	26	ImageNet-1k
31	ResNet	RN101	CNN	SUP	2048	224	45	ImageNet-1k
32	ConvNeXt	ConvNeXt-T	CNN	SUP	768	384	50	ImageNet-21k
33	ConvNeXt	ConvNeXt-B	CNN	SUP	1024	384	89	ImageNet-21k
34	ConvNeXt	ConvNeXt-L	CNN	SUP	1536	384	198	ImageNet-21k
35	ConvNeXt	ConvNeXt-XL	CNN	SUP	2048	384	350	ImageNet-21k
36	DINO	ViT-S/16	Transformer	SSL	384	224	21	ImageNet-1k
37	DINO	ViT-S/8	Transformer	SSL	384	224	21	ImageNet-1k
38	DINO	ViT-B/16	Transformer	SSL	768	224	85	ImageNet-1k
39	DINO	ViT-B/8	Transformer	SSL	768	224	85	ImageNet-1k
40	DINO	RN50	CNN	SSL	2048	224	23	ImageNet-1k
41	DINOv2	ViT-S/14 reg	Transformer	SSL	384	224	21	LVD-142M
42	DINOv2	ViT-B/14 reg	Transformer	SSL	768	224	86	LVD-142M
43	DINOv2	ViT-L/14 reg	Transformer	SSL	1024	224	300	LVD-142M
44	DINOv2	ViT-g/14 reg	Transformer	SSL	1536	224	1100	LVD-142M
45	MoCo v3	ViT-S	Transformer	SSL	384	224	22	ImageNet-1k
46	MoCo v3	ViT-B	Transformer	SSL	768	224	86	ImageNet-1k
47	MoCo v3	RN50	CNN	SSL	2048	224	26	ImageNet-1k