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 [2] both without heavy augmentation (default OpenCLIP setup) and with DINOv2 [5] style color augmentations.[1] We train a ViT-B/32 CVL model both from scratch and finetuned from the LAION-2B checkpoint on the YFCC15M dataset [6]. We follow the default training hyperparameters from OpenCLIP, which is training for 32 epochs with a batch size of $32k$ using the AdamW [4] 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, as some CVL models, like SigLIP [7], encode less metadata information, 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 [3]. 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 $r \sim \text{Uniform}[-20, 20]$. [2]
- **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-

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

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

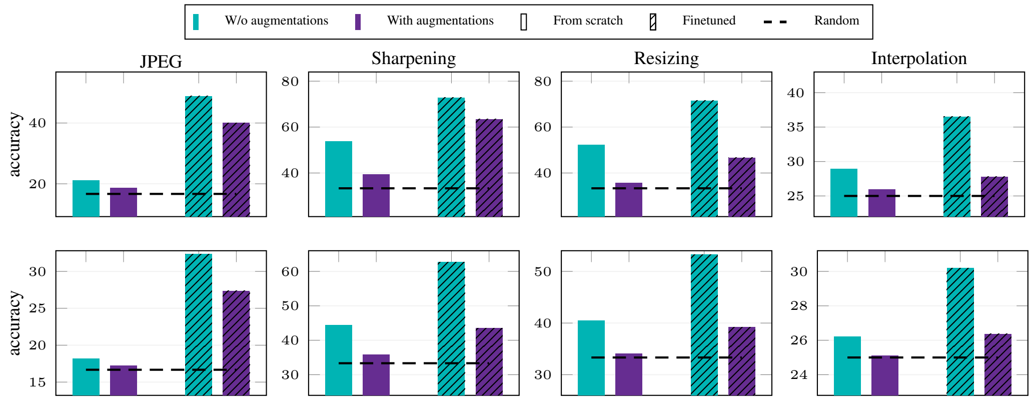

Figure A. **Image processing-based label prediction using a CVL model trained with and without augmentations.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders on ImageNet (top) and iNaturalist (bottom) datasets. CVL model trained both from scratch and finetuned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

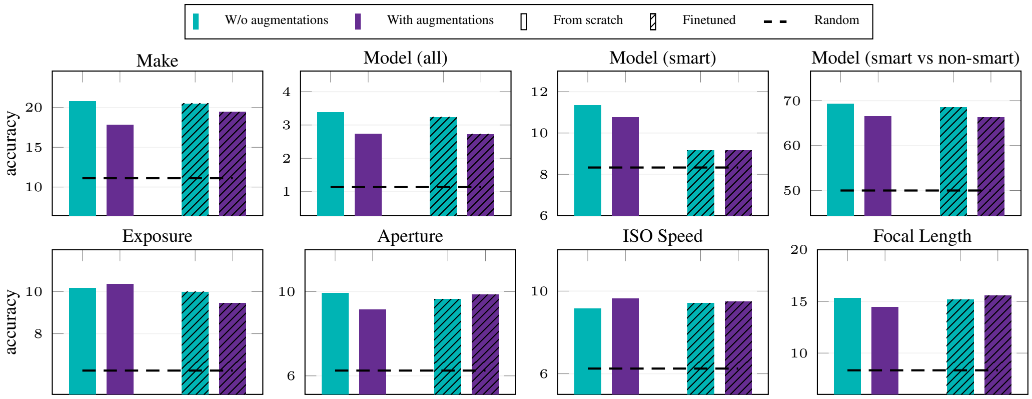

Figure B. **Image acquisition-based label prediction using a CVL model trained with and without augmentations.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders with images masked at 90% on the FlickrExif dataset. CVL model trained both from scratch and finetuned starting from a LAION-2B checkpoint.

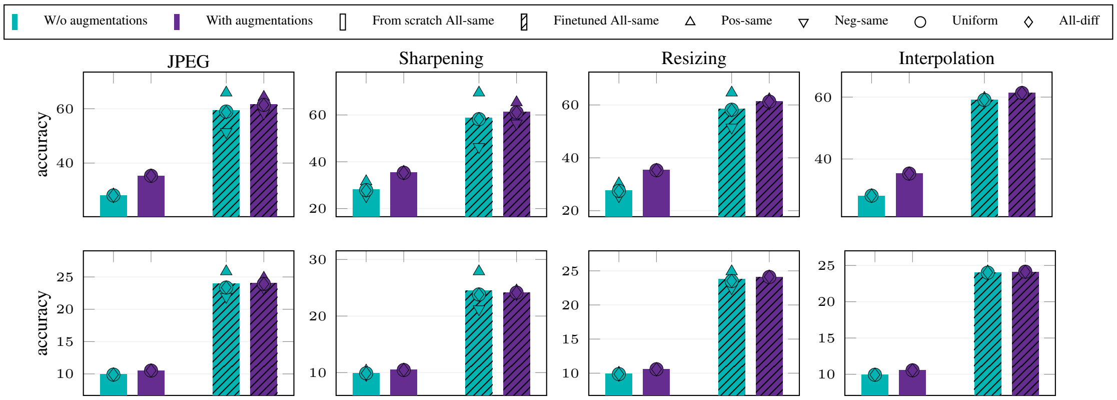

Figure C. **Impact of image processing parameters on semantics for a CVL model trained with and without augmentations.** Semantic label prediction accuracy on ImageNet (top) and iNaturalist (bottom) datasets in five different setups. *All-same (baseline)*: all training and test images share the same 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 10D
Canon EOS 20D
Canon EOS 300D DIGITAL
Canon EOS 30D
Canon EOS 350D DIGITAL
Canon EOS 40D
Canon EOS 450D
Canon EOS 50D
Canon EOS 5D
Canon EOS 5D Mark II
Canon EOS 5D Mark III
Canon EOS 5D Mark IV
Canon EOS 600D
Canon EOS 60D
Canon EOS 6D
Canon EOS 6D Mark II
Canon EOS 70D
Canon EOS 7D
Canon EOS 7D Mark II
Canon EOS 80D
Canon EOS 90D
Canon EOS DIGITAL REBEL
Canon EOS DIGITAL REBEL XT
Canon EOS R
Canon EOS R5
Canon EOS R6
Canon EOS REBEL T3i
Canon EOS Rebel T6
Canon EOS-1D X
Canon EOS-1D X Mark II
E-M1MarkII
E5700
E990
ILCE-6000
ILCE-6400
ILCE-7
ILCE-7M3
ILCE-7RM2
ILCE-7RM3
Kodak CLAS Digital Film Scanner / HR200
NIKON D100
NIKON D200
NIKON D300
NIKON D3100
NIKON D3200
NIKON D40
NIKON D5
NIKON D50
NIKON D500
NIKON D5000
NIKON D5100
NIKON D5200
NIKON D5300
NIKON D5500
NIKON D5600
NIKON D600
NIKON D610
NIKON D70
NIKON D700
NIKON D7000
NIKON D7100
NIKON D7200
NIKON D750
NIKON D7500
NIKON D80
NIKON D800
NIKON D810
NIKON D850
NIKON D90
NIKON Z 6
NIKON Z 6_2
NIKON Z 9
X-T2
X-T3
X-T4
iPhone 11
iPhone 11 Pro Max
iPhone 12 Pro
iPhone 12 Pro Max
iPhone 13 Pro
iPhone 6
iPhone 6s
iPhone 7
iPhone 7 Plus
iPhone X
iPhone XR
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.

D. Visual encoders

Our visual encoders are acquired from the following repositories: OpenAI,[3] OpenCLIP,[4] timm,[5] and FAIR.[6] 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.

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

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

[3] https://github.com/OPENAI
[4] https://github.com/mlfoundations/open_clip
[5] https://github.com/huggingface/pytorch-image-models
[6] https://github.com/facebookresearch/moco-v3

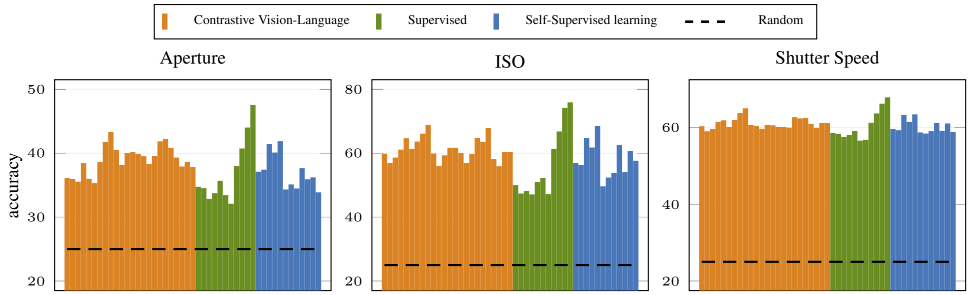

Figure D. **Image acquisition-based label prediction on ImageNet-ES.** Classification accuracy using a linear classifier on embeddings of frozen visual encoders with no masking. Ordering is according to Tab. C.

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.

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. 3

[2] 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. 1

[3] 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

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

[5] 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. 1

[6] 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

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

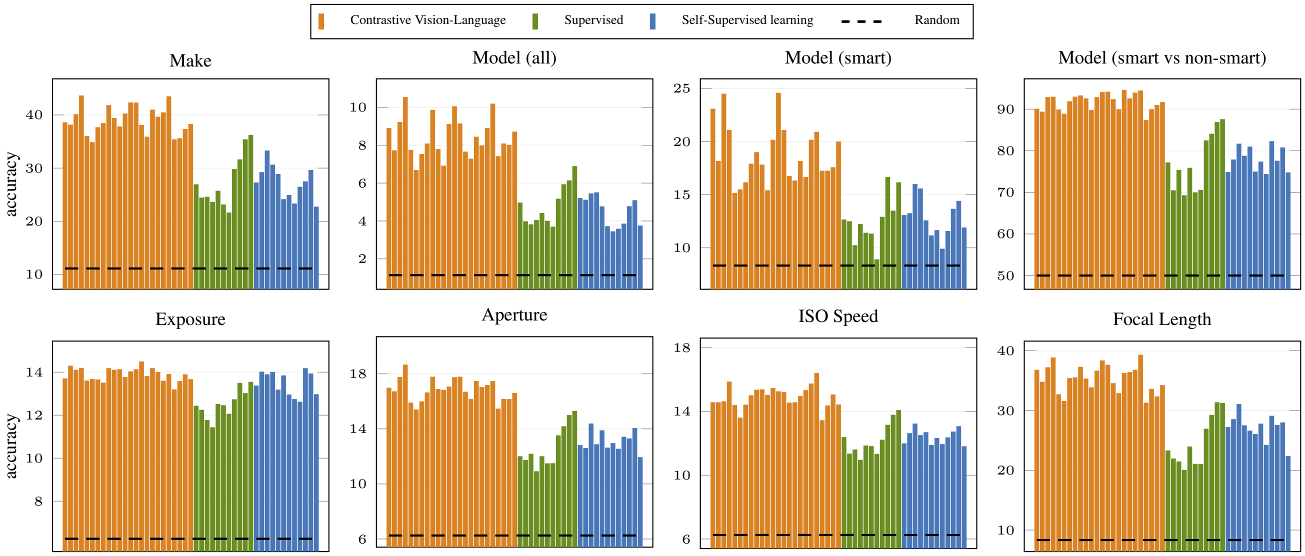

Figure E. **Image acquisition-based label prediction on FlickrExif without masking.** Classification accuracy using a linear classifier. Ordering is according to Tab. C.

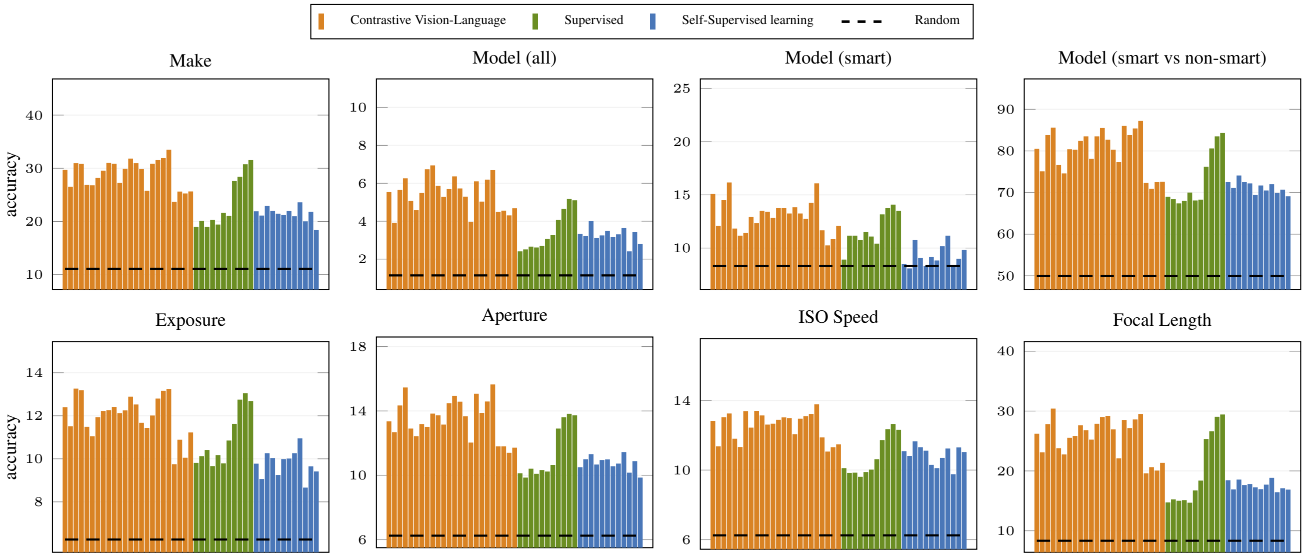

Figure F. **Image acquisition-based label prediction on FlickrExif with a** 75% **masking ratio.** Classification accuracy using a linear classifier. Ordering is according to Tab. C.

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