A. Linear-probe evaluation

We provide additional details for linear probe experiments presented in this paper, including the list of the datasets and models used for evaluation.

A.1. Datasets

We use the 12 datasets from the well-studied evaluation suite introduced by (Kornblith et al., 2019) and add 15 additional datasets in order to assess the performance of models on a wider variety of distributions and tasks. These datasets include MNIST, the Facial Expression Recognition 2013 dataset (Goodfellow et al., 2015), STL-10 (Coates et al., 2011), EuroSAT (Helber et al., 2019), the NWPU-RESISC45 dataset (Cheng et al., 2017), the German Traffic Sign Recognition Benchmark (GTSRB) dataset (Stallkamp et al., 2011), the KITTI dataset (Geiger et al., 2012), PatchCamelyon (Veeling et al., 2018), the UCF101 action recognition dataset (Soomro et al., 2012), Kinetics 700 (Carreira et al., 2019), 2,500 random samples of the CLEVR dataset (Johnson et al., 2017), the Hateful Memes dataset (Kiela et al., 2020), and the ImageNet-1k dataset (Deng et al., 2012). For the two video datasets (UCF101 and Kinetics700), we use the middle frame of each video clip as the input image. STL-10 and UCF101 have multiple predefined train/validation/test splits, 10 and 3 respectively, and we report the average over all splits. Details on each dataset and the corresponding evaluation metrics are provided in Table 2.

Additionally, we created two datasets that we call Country211 and Rendered SST2. The Country211 dataset is designed to assess the geolocation capability of visual representations. We filtered the YFCC100m dataset (Thomee et al., 2016) to find 211 countries (defined as having an ISO-3166 country code) that have at least 300 photos with GPS coordinates, and we built a balanced dataset with 211 categories, by sampling 200 photos for training and 100 photos for testing, for each country.

The Rendered SST2 dataset is designed to measure the optical character recognition capability of visual representations. To do so, we used the sentences from the Stanford Sentiment Treebank dataset (Socher et al., 2013b) and rendered them into images, with black texts on a white background, in a 448×448 resolution. Two example images from this dataset are shown in Figure 8.

A.2. Models

In combination with the datasets listed above, we evaluate the following series of models using linear probes.

LM RN50 This is a multimodal model that uses an autoregressive loss instead of a contrastive loss, while using the ResNet-50 architecture as in the smallest contrastive model. To do so, the output from the CNN is projected into four tokens, which are then fed as a prefix to a language model autoregressively predicting the text tokens. Apart from the training objective, the model was trained on the same dataset for the same number of epochs as other CLIP models.

CLIP-RN Five ResNet-based contrastive CLIP models are included. As discussed in the paper, the first two models follow ResNet-50 and ResNet-101, and we use EfficientNet-style (Tan & Le, 2019) scaling for the next three models which simultaneously scale the model width, the number of layers, and the input resolution to obtain models with roughly 4x, 16x, and 64x computation.

CLIP-ViT We include four CLIP models that use the Vision Transformer (Dosovitskiy et al., 2020) architecture as the image encoder. We include three models trained on 224-by-224 pixel images: ViT-B/32, ViT-B/16, ViT-L/14, and the ViT-L/14 model fine-tuned on 336-by-336 pixel input images.

EfficietNet We use the nine models (B0-B8) from the original EfficientNet paper (Tan & Le, 2019), as well as the noisy-student variants (B0-B7, L2-475, and L2-800) (Tan & Le, 2019). The largest models (L2-475 and L2-800) take the input resolutions of 475x475 and 800x800 pixels, respectively.

Instagram-pretrained ResNeXt We use the four models (32x8d, 32x16d, 32x32d, 32x48d) released by (Mahajan et al., 2018), as well as their two FixRes variants which use higher input resolutions (Touvron et al., 2019).

Big Transfer (BiT) We use BiT-S and BiT-M models (Kolesnikov et al., 2019), trained on the ImageNet-1k and ImageNet-21k datasets. The model weights for BiT-L is not publicly available.

Vision Transformer (ViT) We also include four ViT (Dosovitskiy et al., 2020) checkpoints pretrained on the ImageNet-21k dataset, namely ViT-B/32, ViT-B/16, ViT-L/16, and ViT-H/14. We note that their best-performing models, trained on the JFT-300M dataset, are not available publicly.

SimCLRv2 The SimCLRv2 (Chen et al., 2020a) project released pre-trained and fine-tuned models in various settings. We use the seven pretrain-only checkpoints with selective kernels.

BYOL We use the recently released model weights of BYOL (Grill et al., 2020), specifically their 50x1 and 200x2

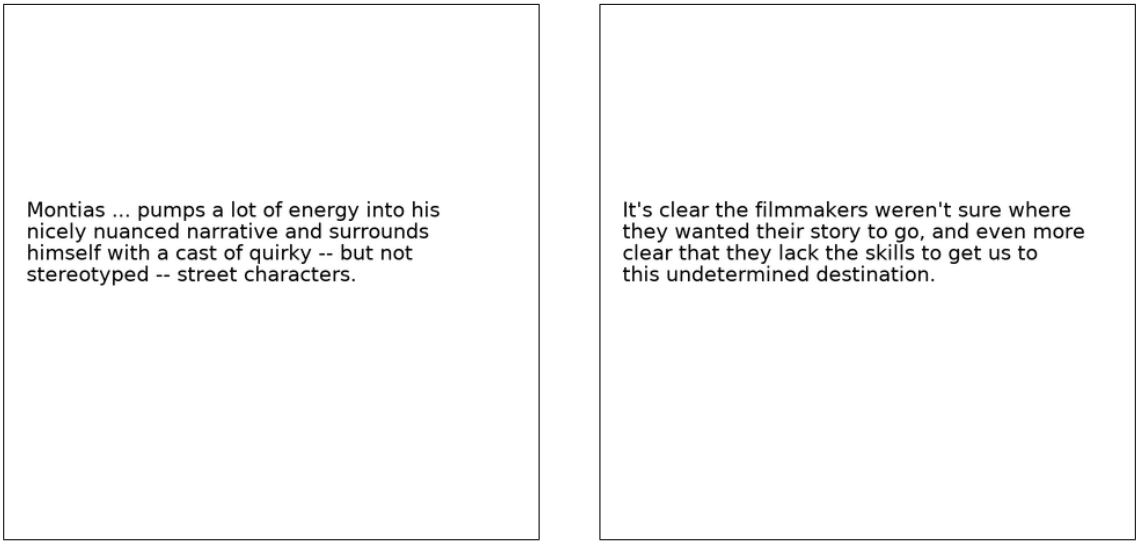

Figure 8. Two example images from the Rendered SST2 dataset

checkpoints.

Momentum Contrast (MoCo) We include the MoCo-v1 (He et al., 2020) and the MoCo-v2 (Chen et al., 2020b) checkpoints.

VirTex We use the pretrained model of VirTex (Desai & Johnson, 2020). We note that VirTex has a similar model design to CLIP-AR but is trained on a 1000x smaller dataset of high-quality captions from MSCOCO.

ResNet We add the original ResNet checkpoints released by (He et al., 2016b), namely ResNet-50, ResNet-101, and ResNet152.

A.3. Evaluation

We use image features taken from the penultimate layer of each model, ignoring any classification layer provided. For CLIP-ViT models, we used the features before the linear projection to the embedding space, which corresponds to `I_f` in Figure 3. We train a logistic regression classifier using scikit-learn's L-BFGS implementation, with maximum 1,000 iterations, and report the corresponding metric for each dataset. We determine the L2 regularization strength λ using a hyperparameter sweep on the validation sets over the range between 10^{-6} and 10^{6}, with 96 logarithmically spaced steps. To save compute required for the sweeps, we perform a parametric binary search that starts with $\lambda = [10^{-6}, 10^{-4}, 10^{-2}, 1, 10^{2}, 10^{4}, 10^{6}]$ and iteratively halves the interval around the peak until it reaches a resolution of 8 steps per decade. The hyperparameter sweeps are performed on a validation split of each dataset. For the datasets that contain a validation split in addition to a test split, we use the provided validation set to perform the hyperparameter search, and for the datasets that do not provide a validation split or have not published labels for the test data, we split the training dataset to perform the hyperparameter search and report the performance on the unused split.

A.4. Results

The individual linear probe scores are provided in Table 3 and plotted in Figure 10. The best-performing CLIP model, using ViT-L/14 archiecture and 336-by-336 pixel images, achieved the state of the art in 21 of the 27 datasets, i.e. included in the Clopper-Pearson 99.5% confidence interval around each dataset's top score. For many datasets, CLIP performs significantly better than other models, demonstrating the advantage of natural language supervision over traditional pre-training approaches based on image classification. See Section 3.3 for more discussions on the linear probe results.

We also visualize per-dataset differences in the performance of the best CLIP model and the best model in our evaluation suite across all 27 datasets in Figure 9. CLIP outperforms the Noisy Student EfficientNet-L2 on 21 of the 27 datasets. CLIP improves the most on tasks which require OCR (SST2 and HatefulMemes), geo-localization and scene recognition (Country211, SUN397), and activity recognition in videos (Kinetics700 and UCF101). In addition CLIP also does much better on fine-grained car and traffic sign recognition (Stanford Cars and GTSRB). This may reflect a problem with overly narrow supervision in ImageNet. A result such as the 14.7% improvement on GTSRB could be indicative of an issue with ImageNet-1K, which has only a single label for all traffic and street signs. This could encourage

Dataset	Classes	Train size	Test size	Evaluation metric
Food-101	102	75,750	25,250	accuracy
CIFAR-10	10	50,000	10,000	accuracy
CIFAR-100	100	50,000	10,000	accuracy
Birdsnap	500	42,283	2,149	accuracy
SUN397	397	19,850	19,850	accuracy
Stanford Cars	196	8,144	8,041	accuracy
FGVC Aircraft	100	6,667	3,333	mean per class
Pascal VOC 2007 Classification	20	5,011	4,952	11-point mAP
Describable Textures	47	3,760	1,880	accuracy
Oxford-IIIT Pets	37	3,680	3,669	mean per class
Caltech-101	102	3,060	6,085	mean-per-class
Oxford Flowers 102	102	2,040	6,149	mean per class
MNIST	10	60,000	10,000	accuracy
Facial Emotion Recognition 2013	8	32,140	3,574	accuracy
STL-10	10	1000	8000	accuracy
EuroSAT	10	10,000	5,000	accuracy
RESISC45	45	3,150	25,200	accuracy
GTSRB	43	26,640	12,630	accuracy
KITTI	4	6,770	711	accuracy
Country211	211	43,200	21,100	accuracy
PatchCamelyon	2	294,912	32,768	accuracy
UCF101	101	9,537	1,794	accuracy
Kinetics700	700	494,801	31,669	mean(top1, top5)
CLEVR Counts	8	2,000	500	accuracy
Hateful Memes	2	8,500	500	ROC AUC
Rendered SST2	2	7,792	1,821	accuracy
ImageNet	1000	1,281,167	50,000	accuracy

Table 2. Datasets examined for linear probes. We note that, for the Birdsnap and Kinetics700 datasets, we used the resources that are available online at the time of this writing.

a supervised representation to collapse intra-class details and hurt accuracy on a fine-grained downstream task. As mentioned, CLIP still underperforms the EfficientNet on several datasets. Unsurprisingly, the dataset that the EfficientNet does best relative to CLIP on is the one it was trained on: ImageNet. The EffcientNet also slightly outperforms CLIP on low-resolution datasets such as CIFAR10 and CIFAR100. We suspect this is at least partly due to the lack of scale-based data augmentation in CLIP. The EfficientNet also does slightly better on PatchCamelyon and CLEVRCounts, datasets where overall performance is still low for both approaches.

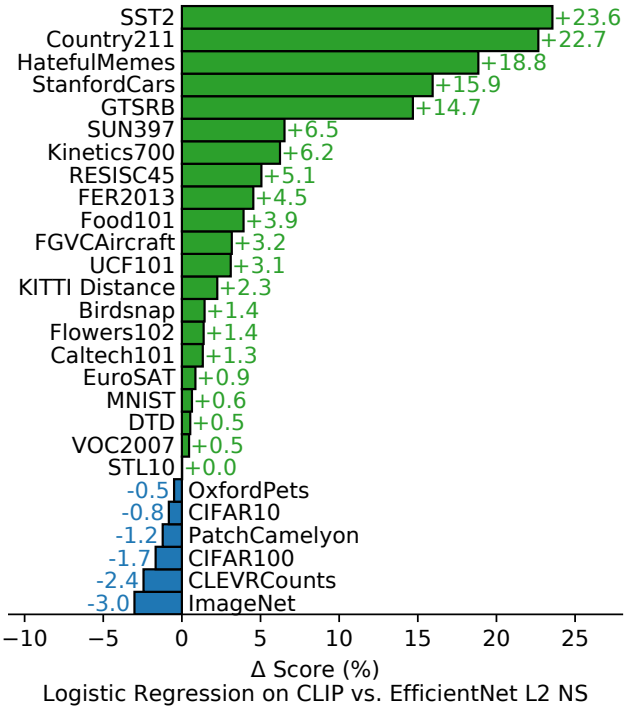

Figure 9. **CLIP's features outperform the features of the best ImageNet model on a wide variety of datasets.** Fitting a linear classifier on CLIP's features outperforms using the Noisy Student EfficientNet-L2 on 21 out of 27 datasets.

		Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech101	Flowers	MNIST	FER2013	STL10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST	ImageNet
LM RN50		81.3	82.8	61.7	44.2	69.6	74.9	44.9	85.5	71.5	82.8	85.5	91.1	96.6	60.1	96.3	93.4	84.0	73.8	70.2	19.0	82.9	76.4	51.9	51.2	65.2	76.8	65.2
CLIP-RN	50	86.4	88.7	70.3	56.4	73.3	78.3	49.1	87.1	76.4	88.2	89.6	96.1	98.3	64.2	97.2	95.2	87.5	82.4	70.2	25.3	82.7	81.6	57.2	53.6	65.7	72.6	73.3
	101	88.9	91.1	73.5	58.6	75.1	84.0	50.7	88.0	76.3	91.0	92.0	96.4	98.4	65.2	98.2	95.9	89.3	82.4	**73.6**	26.6	82.8	84.0	60.3	50.3	68.2	73.3	75.7
	50x4	91.3	90.5	73.0	65.7	77.0	85.9	57.3	88.4	79.5	91.9	92.5	97.8	98.5	68.1	98.1	96.4	89.7	85.5	59.4	30.3	83.0	85.7	62.6	52.5	68.0	76.6	78.2
	50x16	93.3	92.2	74.9	72.8	79.2	88.7	62.7	**89.0**	79.1	93.5	93.7	98.3	**98.9**	68.7	99.0	97.0	91.4	89.0	69.2	34.8	83.5	88.0	66.3	53.8	71.1	**80.0**	81.5
	50x64	94.8	94.1	78.6	77.2	81.1	90.5	67.7	**88.9**	**82.0**	94.5	95.4	98.9	**98.9**	**71.3**	99.3	97.1	92.8	90.2	69.2	40.7	83.7	89.5	69.1	55.0	**75.0**	**81.2**	83.6
CLIP-ViT	B/32	88.8	95.1	80.5	58.5	76.6	81.8	52.0	87.7	76.5	90.0	93.0	96.9	**99.0**	69.2	98.6	97.0	90.5	85.3	66.2	27.8	83.9	85.5	61.7	52.1	66.7	70.8	76.1
	B/16	92.8	96.2	83.1	67.8	78.4	86.7	59.5	**89.2**	79.2	93.1	94.7	98.1	**99.0**	69.5	99.2	97.1	92.7	86.6	67.8	33.3	83.5	88.4	66.1	**57.1**	70.3	75.5	80.2
	L/14	95.2	98.0	87.5	77.0	**81.8**	**90.9**	69.4	**89.6**	**82.1**	**95.1**	**96.5**	99.2	**99.2**	**72.2**	**99.8**	**98.2**	94.1	**92.5**	64.7	42.9	85.8	**91.5**	72.0	**57.8**	**76.2**	**80.8**	83.9
	L/14-336px	**95.9**	97.9	87.4	**79.9**	**82.2**	**91.5**	**71.6**	**89.9**	**83.0**	**95.1**	**96.0**	99.2	**99.2**	**72.9**	**99.7**	**98.1**	**94.9**	**92.4**	69.2	**46.4**	85.6	**92.0**	**73.0**	**60.3**	**77.3**	**80.5**	85.4
EfficientNet	B0	74.3	92.5	76.5	59.7	62.0	62.5	55.7	84.4	71.2	93.0	93.3	91.7	98.2	57.2	97.8	97.3	85.5	80.0	**73.8**	12.4	83.1	74.4	47.6	47.9	55.7	53.4	76.9
	B1	74.2	93.2	77.2	61.3	62.6	62.5	56.1	84.7	74.2	93.4	93.6	92.4	98.3	57.0	97.9	96.8	84.5	75.9	**75.5**	12.5	82.7	74.7	48.5	44.3	54.5	54.4	78.6
	B2	75.8	93.6	77.9	64.4	64.0	63.2	57.0	85.3	73.5	93.9	93.5	92.9	98.5	56.6	98.1	96.9	84.4	76.4	**73.1**	12.6	84.3	75.1	49.4	42.6	55.4	55.2	79.7
	B3	77.4	94.0	78.0	66.5	64.4	66.0	59.3	85.8	73.1	94.1	93.7	93.3	98.5	57.1	98.6	97.3	85.0	75.8	**76.1**	13.4	83.3	78.1	50.9	45.1	53.8	54.8	81.0
	B4	79.7	94.1	78.7	70.1	65.4	66.4	60.4	86.5	73.4	94.7	93.5	93.2	98.8	57.9	98.9	96.8	85.0	78.3	**72.3**	13.9	83.1	79.1	52.5	46.5	54.4	55.4	82.9
	B5	81.5	93.6	77.9	72.4	67.1	72.7	68.9	86.7	73.9	**95.0**	94.7	94.5	98.4	58.5	99.1	96.8	86.0	78.5	69.6	14.9	84.7	80.9	54.5	46.6	53.3	56.3	83.7
	B6	82.4	94.0	78.0	73.5	65.8	71.1	68.2	87.6	73.9	**95.0**	94.1	93.7	98.4	60.2	99.0	96.8	85.4	78.1	**72.7**	15.3	84.2	80.0	54.1	51.1	53.3	57.0	84.0
	B7	84.5	94.9	80.1	74.7	69.0	77.1	**72.3**	87.2	76.8	**95.2**	94.7	95.9	98.6	61.3	99.3	96.3	86.8	80.8	**75.8**	16.4	85.2	81.9	56.8	51.9	54.4	57.8	84.8
	B8	84.5	95.0	80.7	75.2	69.6	76.8	**71.5**	87.4	77.1	**94.9**	95.2	96.3	98.6	61.4	99.4	97.0	87.4	80.4	70.9	17.4	85.2	82.4	57.7	51.4	51.7	55.8	85.3
EfficientNet Noisy Student	B0	78.1	94.0	78.6	63.5	65.5	57.2	53.7	85.6	75.6	93.8	93.1	94.5	98.1	55.6	98.6	97.0	84.3	74.0	71.6	14.0	83.1	76.7	51.7	47.3	55.7	55.0	78.5
	B1	80.4	95.1	80.2	66.6	67.6	59.6	53.7	86.2	77.0	94.6	94.4	95.1	98.0	56.1	98.9	96.9	84.3	73.1	67.1	14.5	83.9	79.9	54.5	46.1	54.3	54.9	81.1
	B2	80.9	95.3	81.3	67.6	67.9	60.9	55.2	86.3	77.7	**95.0**	94.7	94.4	98.0	55.5	98.9	97.3	84.6	71.7	70.0	14.6	82.9	80.1	55.1	46.1	54.1	55.3	82.2
	B3	82.6	95.9	82.1	68.6	68.8	60.6	55.4	86.5	77.2	**95.0**	94.8	95.2	98.1	56.0	99.3	96.5	85.0	70.5	69.5	15.1	83.1	81.8	56.8	45.1	55.7	52.0	83.8
	B4	85.2	95.6	81.0	72.5	69.7	56.1	52.6	87.0	78.7	**94.8**	95.2	95.3	98.2	56.0	99.4	95.3	84.8	61.9	64.8	16.0	82.8	83.4	59.8	43.2	55.3	53.0	85.4
	B5	87.6	96.3	82.4	75.3	71.6	64.7	64.8	87.8	79.6	**95.5**	95.6	96.6	98.8	60.9	99.5	96.1	87.0	68.5	**73.7**	16.4	83.5	86.4	61.6	46.3	53.4	55.8	85.8
	B6	87.3	97.0	83.9	75.8	71.4	67.6	65.6	87.3	78.5	**95.2**	**96.4**	97.2	98.6	61.9	**99.7**	96.6	86.1	70.7	**72.4**	17.6	84.2	85.5	61.0	49.6	54.6	55.7	86.4
	B7	88.4	96.0	82.0	76.9	72.6	72.2	**71.2**	88.1	**80.5**	**95.5**	95.5	96.6	98.5	62.7	99.5	96.2	88.5	73.4	**73.0**	18.5	83.8	86.6	63.2	50.5	57.2	56.7	87.0
	L2-475	91.6	**99.0**	**91.0**	74.8	76.4	75.1	66.8	**89.5**	**81.9**	**95.6**	**96.5**	97.7	**98.9**	67.5	**99.7**	97.0	89.5	73.4	68.9	22.2	86.3	89.4	68.2	**58.3**	58.6	55.2	**88.3**
	L2-800	92.0	**98.7**	89.0	**78.5**	75.7	75.5	68.4	**89.4**	**82.5**	**95.6**	94.7	97.9	98.5	68.4	**99.7**	97.2	89.9	77.7	66.9	23.7	**86.8**	88.9	66.7	**62.7**	58.4	56.9	**88.4**
Instagram	32x8d	84.8	95.9	80.9	63.8	69.0	74.2	56.0	88.0	75.4	**95.4**	93.9	91.7	97.4	60.7	99.3	95.7	82.1	72.3	69.2	16.7	82.3	80.1	56.8	42.2	53.3	55.2	83.3
	32x16d	85.7	96.5	80.9	64.8	70.5	77.5	56.7	87.9	76.2	**95.6**	94.9	92.5	97.4	61.6	99.4	95.5	82.8	73.8	66.1	17.5	83.4	81.1	58.2	41.3	54.2	56.1	84.4
	32x32d	86.7	96.8	82.7	67.1	71.5	77.5	55.4	88.3	78.5	**95.8**	95.3	94.4	97.9	62.4	99.4	95.7	85.4	71.2	66.8	18.0	83.7	82.1	58.8	39.7	55.3	56.7	85.0
	32x48d	86.9	96.8	83.4	65.9	72.2	76.6	53.2	88.0	77.2	**95.5**	95.8	93.6	98.1	63.7	99.5	95.3	85.4	73.0	67.2	18.5	82.7	82.8	59.2	41.3	55.5	56.7	85.2
	FixRes-v1	88.5	95.7	81.1	67.4	72.9	80.5	57.6	88.0	77.9	**95.8**	**96.1**	94.5	97.9	62.2	99.5	96.2	86.6	76.5	64.8	19.3	82.5	83.4	59.8	43.5	56.6	59.0	86.0
	FixRes-v2	88.5	95.7	81.1	67.3	72.9	80.7	57.5	88.0	77.9	**95.0**	**96.0**	94.5	98.0	62.1	99.5	96.5	86.6	76.3	64.8	19.5	82.3	83.5	59.8	44.2	56.6	59.0	86.0
BiT-S	R50x1	72.5	91.7	74.8	57.7	61.1	53.5	52.5	83.7	72.4	92.3	91.2	92.0	98.4	56.1	97.1	97.4	85.0	70.0	66.0	12.5	83.0	72.3	47.5	48.3	54.1	55.3	75.2
	R50x3	75.1	93.7	79.0	61.1	63.7	55.2	54.1	84.8	74.6	92.5	91.6	92.8	98.8	58.7	97.7	**97.8**	86.4	73.1	**73.8**	14.0	84.2	76.4	50.0	49.2	54.7	54.2	77.2
	R101x1	73.5	92.8	77.4	58.4	61.3	54.0	52.4	84.4	73.5	92.5	91.8	90.6	98.3	56.5	97.6	97.3	84.6	69.4	68.9	12.6	82.0	73.5	48.6	45.4	52.6	55.5	76.0
	R101x3	74.7	93.9	79.8	57.8	62.9	54.7	53.3	84.7	75.5	92.3	91.2	92.6	98.8	59.7	98.0	**98.0**	85.5	71.8	60.2	14.1	83.1	75.9	50.4	49.7	54.1	54.6	77.4
	R152x2	74.9	94.3	79.7	58.7	62.7	55.9	53.6	85.3	74.9	93.0	92.0	91.7	98.6	58.3	97.6	**97.8**	86.2	71.8	71.6	13.9	84.1	76.2	49.9	48.2	53.8	55.9	77.1
	R152x4	74.7	94.2	79.2	57.8	62.9	51.2	50.8	85.4	75.4	93.1	91.2	91.4	**98.9**	61.4	98.0	**98.0**	85.5	72.8	67.9	14.9	83.1	76.0	50.3	42.9	53.6	56.0	78.5
BiT-M	R50x1	83.3	94.9	82.2	70.9	69.9	59.0	55.6	86.8	77.3	91.5	93.9	**99.4**	98.0	60.6	98.7	97.5	87.4	68.6	68.2	16.6	82.5	79.4	53.2	49.4	54.5	53.4	76.7
	R50x3	86.9	96.7	86.2	75.7	74.6	60.6	54.2	87.7	78.5	93.2	95.3	**99.4**	98.6	64.6	99.5	**98.0**	88.1	69.9	59.6	19.6	83.4	83.5	57.8	51.3	55.8	55.6	80.7
	R101x1	85.5	95.7	84.4	73.0	72.5	59.8	55.0	87.3	78.1	92.2	95.0	**99.5**	98.1	62.5	99.2	**97.6**	87.8	68.7	67.7	18.0	84.0	82.3	55.9	53.4	54.8	53.1	79.4
	R101x3	87.2	97.4	87.5	72.4	75.0	57.4	47.4	87.5	79.6	93.2	95.4	**99.6**	98.6	64.3	99.5	**98.2**	87.7	68.8	64.1	20.7	80.4	84.0	58.7	52.6	54.9	54.3	81.2
	R152x2	88.0	97.5	87.8	75.8	75.9	61.5	55.3	88.1	79.8	93.6	**95.9**	**99.5**	98.5	64.3	99.6	**97.9**	89.0	70.0	70.3	20.7	82.6	85.5	59.6	50.8	54.9	55.1	81.9
	R152x4	87.2	97.6	88.2	72.4	75.0	49.1	43.4	87.1	79.9	92.4	95.4	99.3	98.5	65.7	**99.6**	**97.8**	87.7	68.2	57.1	20.6	80.4	84.6	59.0	49.7	57.2	55.1	81.5
ViT	B/32	81.8	96.7	86.3	65.2	70.7	49.1	42.7	85.3	73.1	90.4	94.5	98.7	97.8	59.0	99.1	96.3	83.0	68.1	65.1	15.7	82.6	79.1	51.7	38.9	57.1	54.6	76.6
	B/16	86.7	96.9	86.4	74.0	74.2	54.7	46.0	86.7	74.3	92.7	94.1	99.2	97.4	61.3	99.5	96.4	84.5	63.1	61.5	17.5	85.4	82.7	56.6	40.0	57.0	56.1	80.9
	L/16	87.4	97.9	89.0	76.5	74.9	62.5	52.2	86.1	75.0	92.9	94.7	99.3	98.0	64.0	**99.7**	96.5	85.7	70.4	58.8	17.7	85.7	84.1	58.0	38.4	58.4	52.8	81.9
	H/14	83.4	95.8	84.5	70.2	69.2	62.3	54.8	84.7	75.4	91.7	93.7	98.9	98.5	62.4	98.5	97.3	87.0	73.9	63.4	15.4	**87.0**	79.4	52.1	41.1	55.9	54.1	75.9
SimCLRv2	R50x1	76.4	93.2	77.9	48.6	64.1	56.3	51.7	84.4	77.0	88.3	91.8	92.9	97.6	59.7	97.9	97.5	85.8	71.1	69.1	15.8	84.8	78.4	51.0	56.2	53.9	53.8	73.8
	R50x3	81.0	95.6	82.4	56.5	67.0	65.6	61.1	85.9	78.8	90.9	94.1	95.4	98.7	62.6	98.8	**97.9**	88.2	78.2	**74.7**	17.6	85.4	82.6	54.6	55.4	54.2	55.2	77.3
	R101x1	77.9	94.8	79.9	51.9	65.2	57.1	52.0	85.4	77.2	90.0	91.6	92.7	97.2	59.4	98.2	96.8	84.6	65.7	70.6	16.1	84.3	78.8	52.4	53.6	55.1	55.7	76.1
	R101x3	82.2	96.4	83.4	57.5	68.2	64.6	60.0	86.2	78.9	91.8	95.0	95.4	98.4	63.0	99.0	**97.9**	88.0	77.5	69.1	18.3	85.5	82.9	55.9	52.2	54.5	56.3	78.8
	R152x1	78.6	95.0	79.9	50.3	65.6	55.6	52.2	85.8	77.3	90.1	92.5	91.8	97.6	59.8	98.6	96.6	84.3	64.8	70.3	16.6	83.9	79.4	53.1	**57.2**	55.8	54.8	76.9
	R152x2	82.3	96.7	83.9	58.1	68.5	64.9	58.7	86.6	79.1	92.2	94.1	96.0	98.2	64.1	99.0	**98.0**	88.1	77.0	69.8	18.4	85.3	82.7	56.2	53.6	56.0	56.5	79.2
	R152x3	83.6	96.8	84.5	60.3	69.1	68.5	63.1	86.7	**80.5**	92.6	94.9	96.3	98.7	65.4	99.2	**98.1**	89.5	78.4	68.5	19.4	85.2	83.5	57.0	54.4	54.6	54.2	80.0
BYOL	50x1	74.0	93.6	79.1	47.6	63.7	61.6	62.3	82.6	77.0	88.3	93.7	94.3	98.7	58.8	97.4	**97.6**	88.2	80.1	71.4	14.1	84.8	77.3	49.3	56.1	53.8	54.4	73.3
	200x2	78.5	96.2	83.3	53.4	68.5	61.7	55.4	86.6	77.4	91.9	95.5	93.9	98.7	62.6	99.0	**97.7**	87.4	77.1	**76.4**	16.4	84.0	82.6	55.1	54.1	52.5	52.4	79.2
MoCo	v1	65.9	85.0	63.1	27.5	52.6	35.9	43.5	75.7	70.0	70.4	78.1	85.4	97.6	54.3	90.1	97.1	82.9	62.6	60.2	12.6	85.7	64.2	40.7	54.7	55.6	53.5	57.2
	v2	72.2	93.4	76.3	39.6	60.2	48.3	51.1	82.6	75.1	84.4	89.9	90.7	98.4	58.3	96.9	97.2	85.4	75.7	**75.4**	13.2	85.6	72.7	47.8	**56.9**	53.9	53.8	69.1
VirTex		57.9	83.9	57.5	17.0	49.8	22.4	34.5	83.8	58.2	53.6	70.6	74.7	98.1	56.5	89.8	94.8	74.1	69.5	71.3	8.7	83.1	61.5	39.9	45.5	53.5	55.8	50.7
ResNet	50	71.3	91.8	74.5	52.7	60.5	49.9	48.5	83.8	72.3	92.4	90.8	90.8	98.3	54.9	97.1	96.7	83.6	70.6	67.1	11.7	82.5	71.2	46.8	43.0	56.5	55.5	74.3
	101	72.7	93.0	77.2	53.7	60.8	50.1	47.0	84.4	71.6	92.3	91.9	90.4	98.5	56.6	97.8	97.1	83.4	72.5	63.6	11.9	83.3	72.7	48.3	43.2	53.0	54.7	75.8
	152	73.7	93.5	78.0	55.1	61.6	52.8	48.4	84.5	71.9	93.0	92.1	89.6	98.2	57.0	98.2	97.0	83.1	70.1	70.2	12.3	82.9	75.3	49.2	42.4	53.2	53.9	77.1

Table 3. Linear probe performance. Scores within the 99.5% Clopper-Pearson confidence interval of each dataset's top score are shown in bold.

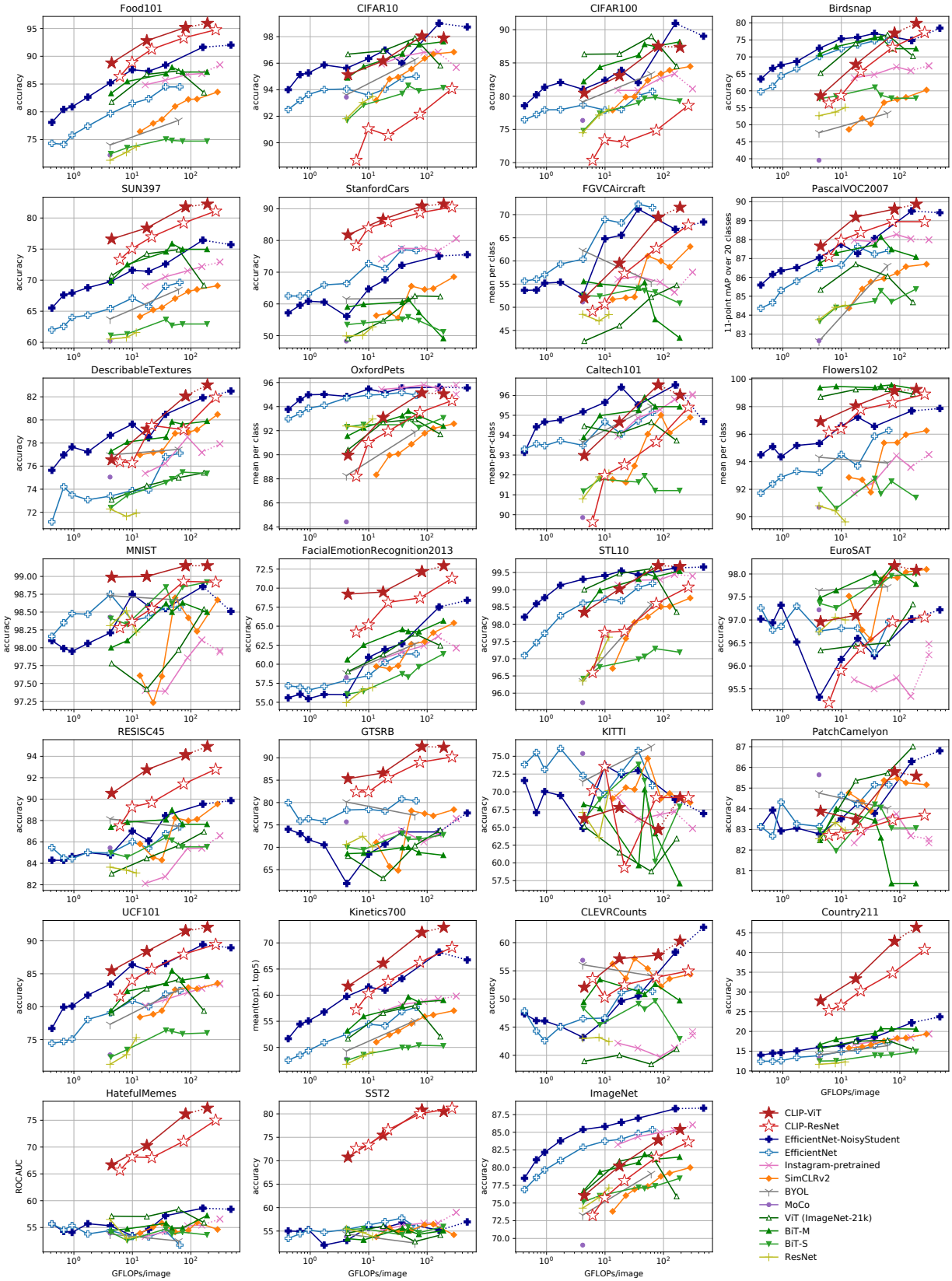

Figure 10. Linear probe performance plotted for each of the 27 datasets, using the data from Table 3.

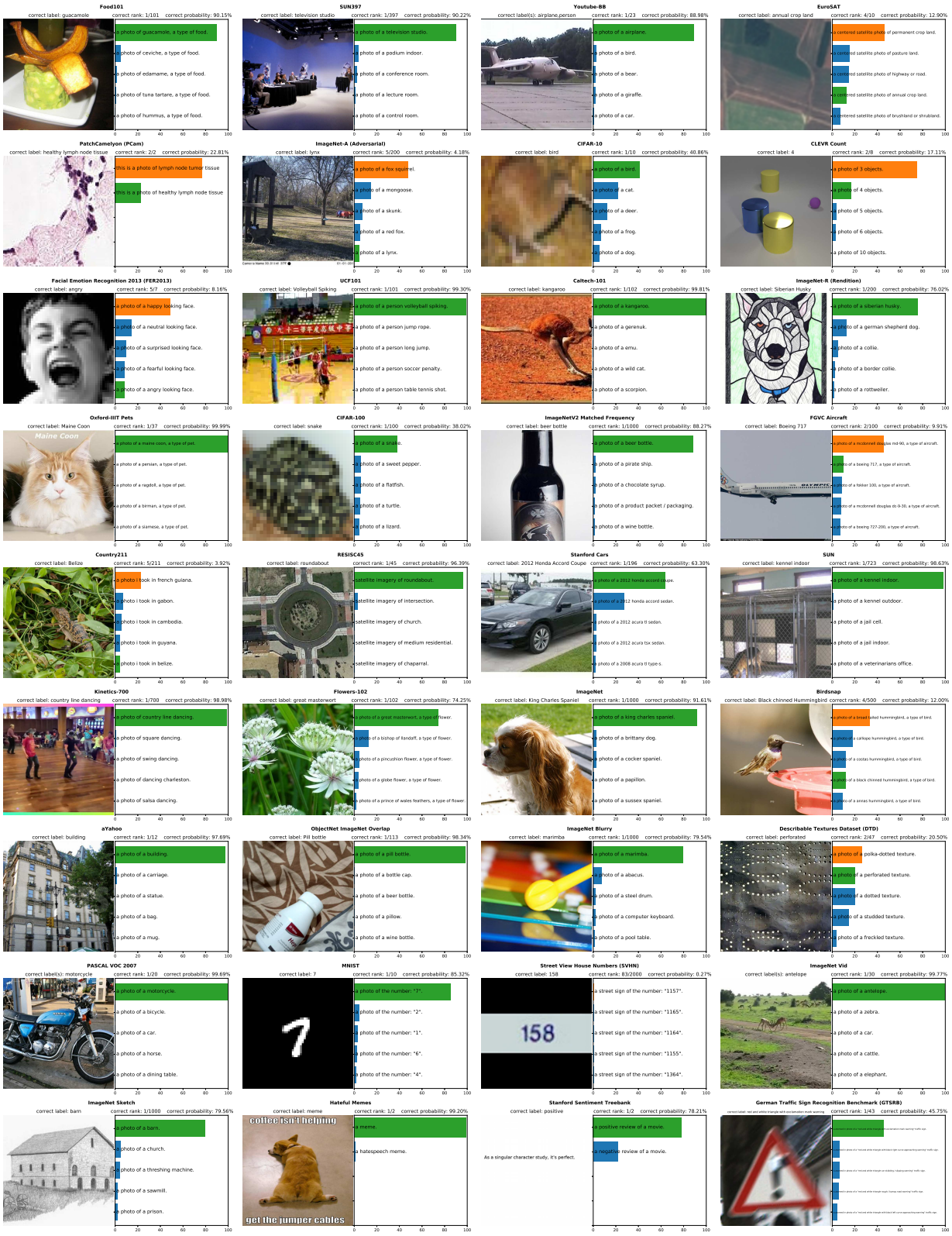

Figure 11. Visualization of predictions from 36 CLIP zero-shot classifiers. All examples are random with the exception of reselecting Hateful Memes to avoid offensive content. The predicted probability of the top 5 classes is shown along with the text used to represent the class. When more than one template is used, the first template is shown. The ground truth label is colored green while an incorrect prediction is colored orange.

		Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Stanford Cars	FGVC Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102	MNIST	FER2013	STL10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCam	UCF101	Kinetics700	CLEVR	HatefulMemes	Rendered SST2	ImageNet
CLIP-ResNet	RN50	81.1	75.6	41.6	32.6	59.6	55.8	19.3	82.1	41.7	85.4	82.1	65.9	66.6	42.2	94.3	41.1	54.2	35.2	42.2	16.1	57.6	63.6	43.5	20.3	59.7	56.9	59.6
	RN101	83.9	81.0	49.0	37.2	59.9	62.3	19.5	82.4	43.9	86.2	85.1	65.7	59.3	45.6	96.7	33.1	58.5	38.3	33.3	16.9	55.2	62.2	46.7	28.1	61.1	64.2	62.2
	RN50x4	86.8	79.2	48.9	41.6	62.7	67.9	24.6	83.0	49.3	88.1	86.0	68.0	75.2	51.1	96.4	35.0	59.2	35.7	26.0	20.2	57.5	65.5	49.0	17.0	58.3	66.6	65.8
	RN50x16	90.5	82.2	54.2	45.9	65.0	72.3	30.3	82.9	52.8	89.7	87.6	71.9	80.0	56.0	97.8	40.3	64.4	39.6	33.9	24.0	62.5	68.7	53.4	17.6	58.9	67.6	70.5
	RN50x64	91.8	86.8	61.3	48.9	66.9	76.0	35.6	83.8	53.4	93.4	90.6	77.3	90.8	61.0	98.3	59.4	69.7	47.9	33.2	29.6	65.0	74.1	56.8	27.5	62.1	70.7	73.6
CLIP-ViT	B/32	84.4	91.3	65.1	37.8	63.2	59.4	21.2	83.1	44.5	87.0	87.9	66.7	51.9	47.3	97.2	49.4	60.3	32.2	39.4	17.8	58.4	64.5	47.8	24.8	57.6	59.6	63.2
	B/16	89.2	91.6	68.7	39.1	65.2	65.6	27.1	83.9	46.0	88.9	89.3	70.4	56.0	52.7	98.2	54.1	65.5	43.3	44.0	23.3	48.1	69.8	52.4	23.4	61.7	59.8	68.6
	L/14	92.9	96.2	77.9	48.3	67.7	77.3	36.1	84.1	55.3	93.5	92.6	78.7	87.2	57.5	99.3	59.9	71.6	50.3	23.1	32.7	58.8	76.2	60.3	24.3	63.3	64.0	75.3
	L/14-336px	93.8	95.7	77.5	49.5	68.4	78.8	37.2	84.3	55.7	93.5	92.8	78.3	88.3	57.7	99.4	59.6	71.7	52.3	21.9	34.9	63.0	76.9	61.3	24.8	63.3	67.9	76.2

Table 4. Zero-shot performance of CLIP models over 27 datasets.

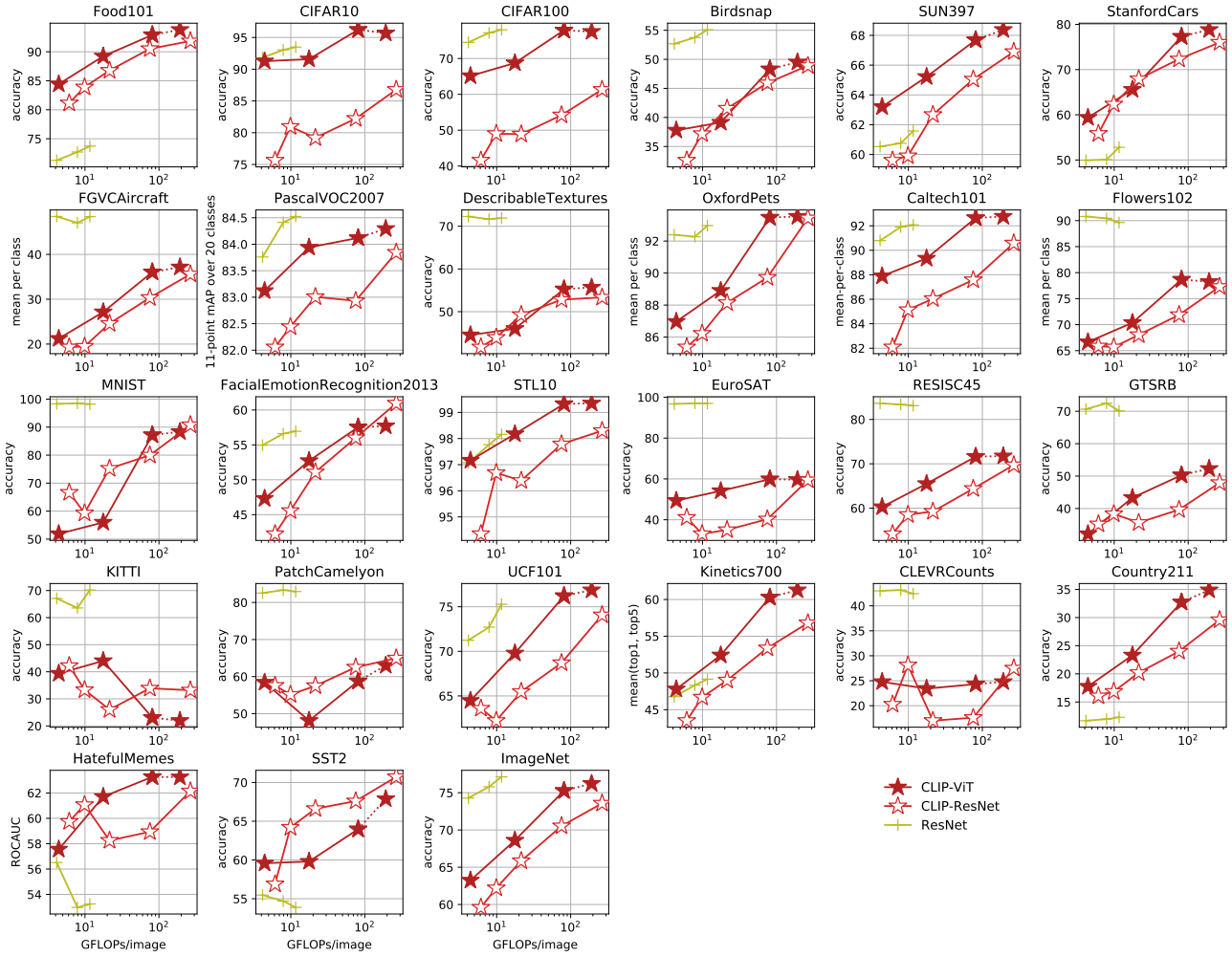

Figure 12. CLIP's zero-shot performance compared to linear-probe ResNet performance

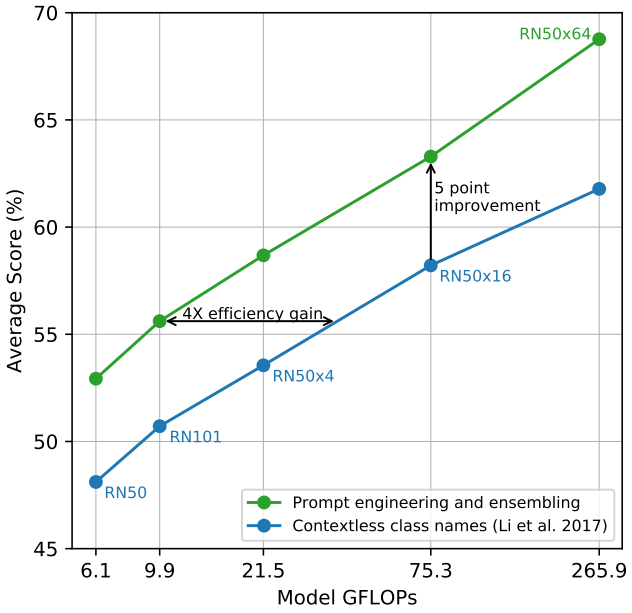

Figure 13. **Prompt engineering and ensembling improve zero-shot performance.** Compared to the baseline of using contextless class names, prompt engineering and ensembling boost zero-shot classification performance by almost 5 points on average across 36 datasets. This improvement is similar to the gain from using 4 times more compute with the baseline zero-shot method but is "free" when amortized over many predictions.

B. Zero-Shot Analysis

To provide a qualitative summary / overview of CLIP's zero-shot performance we visualize a randomly selected prediction for 36 different zero-shot CLIP classifiers in Figure 11. In addition, Table 4 and Figure 12 show the individual zero-shot performance scores for each dataset. In the following, we describe additional details of our zero-shot results.

Most standard image classification datasets treat the information naming or describing classes which enables natural language based zero-shot transfer as an afterthought. The vast majority of datasets annotate images with just a numeric id of the label and contain a file mapping these ids back to their names in English. Some datasets, such as Flowers102 and GTSRB, don't appear to include this mapping at all in their released versions preventing zero-shot transfer entirely. For many datasets, we observed these labels may be chosen somewhat haphazardly and do not anticipate issues related to zero-shot transfer which relies on task description in order to transfer successfully.

A common issue is polysemy. When the name of a class is the only information provided to CLIP's text encoder it is unable to differentiate which word sense is meant due to the lack of context. In some cases multiple meanings of the same word might be included as different classes in the same dataset! This happens in ImageNet which contains both construction cranes and cranes that fly. Another example is found in classes of the Oxford-IIIT Pet dataset where the word boxer is, from context, clearly referring to a breed of dog, but to a text encoder lacking context could just as likely refer to a type of athlete.

Another issue we encountered is that it's relatively rare in our pre-training dataset for the text paired with the image to be just a single word. Usually the text is a full sentence describing the image in some way. To help bridge this distribution gap, we found that using the prompt template "`A photo of a {label}.`" to be a good default that helps specify the text is about the content of the image. This often improves performance over the baseline of using only the label text. For instance, just using this prompt improves accuracy on ImageNet by 1.3%.

Similar to the "prompt engineering" discussion around GPT-3 (Brown et al., 2020; Gao et al., 2020), we have also observed that zero-shot performance can be significantly improved by customizing the prompt text to each task. A few, non exhaustive, examples follow. We found on several fine-grained image classification datasets that it helped to specify the category. For example on Oxford-IIIT Pets, using "`A photo of a {label}, a type of pet.`" to help provide context worked well. Likewise, on Food101 specifying *a type of food* and on FGVC Aircraft *a type of aircraft* helped too. For OCR datasets, we found that putting quotes around the text or number to be recognized improved performance. Finally, we found that on satellite image classification datasets it helped to specify that the images were of this form and we use variants of "`a satellite photo of a {label}.`".

We also experimented with ensembling over multiple zero-shot classifiers as another way of improving performance. These classifiers are computed by using different context prompts such as '`A photo of a big {label}`" and "`A photo of a small {label}`". We construct the ensemble over the embedding space instead of probability space. This allows us to cache a single set of averaged text embeddings so that the compute cost of the ensemble is the same as using a single classifier when amortized over many predictions. We've observed ensembling across many generated zero-shot classifiers to reliably improve performance and use it for the majority of datasets. On ImageNet, we ensemble 80 different context prompts and this improves performance by an additional 3.5% over the single default prompt discussed above. When considered together, prompt engineering and ensembling improve ImageNet accuracy by almost 5%. In Figure 13 we visualize how prompt engineering and ensembling change the performance of a set of CLIP models compared to the contextless baseline approach of directly embedding the class name as done in Li et al.

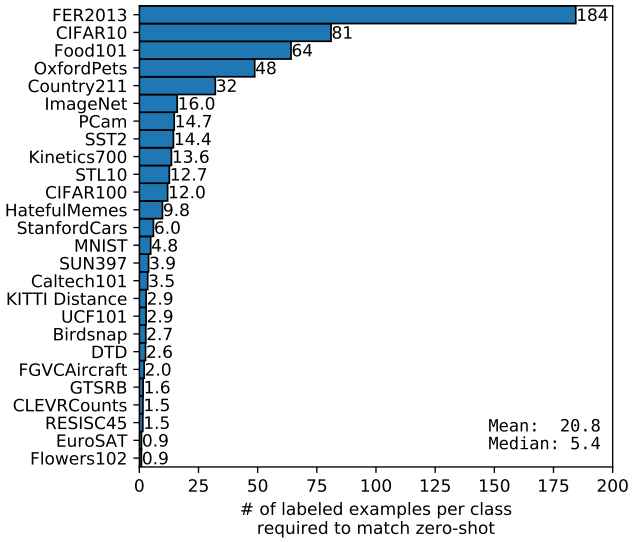

Figure 14. **The data efficiency of zero-shot transfer varies widely.** Calculating the number of labeled examples per class a linear classifier on the same CLIP feature space requires to match the performance of the zero-shot classifier contextualizes the effectiveness of zero-shot transfer. Values are estimated based on log-linear interpolation of 1, 2, 4, 8, 16-shot and fully supervised results. Performance varies widely from still underperforming a one-shot classifier on two datasets to matching an estimated 184 labeled examples per class.

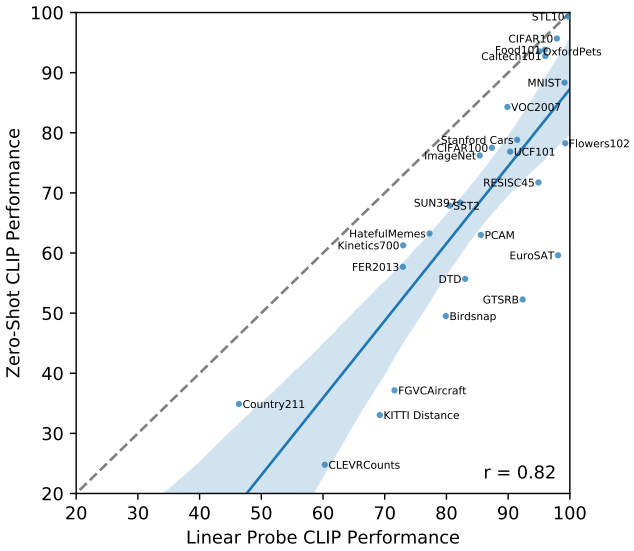

Figure 15. **Zero-shot performance is correlated with linear probe performance but still mostly sub-optimal.** Comparing zero-shot and linear probe performance across datasets shows a strong correlation with zero-shot performance mostly shifted 10 to 25 points lower. On only 5 datasets does zero-shot performance approach linear probe performance ($\leq$3 point difference).

(2017).

In addition to studying the average performance of zero-shot CLIP and few-shot logistic regression in the main body, we also examine performance on individual datasets. In Figure 14, we show estimates for the number of labeled examples per class that a logistic regression classifier on the same feature space requires to match the performance of zero-shot CLIP. Since zero-shot CLIP is also a linear classifier, this estimates the effective data efficiency of zero-shot transfer in this setting. In order to avoid training thousands of linear classifiers, we estimate the effective data efficiency based on a log-linear interpolation of the performance of a 1, 2, 4, 8, 16-shot (when possible), and a fully supervised linear classifier trained on each dataset. We find that zero-shot transfer can have widely varying efficiency per dataset from less than 1 labeled example per class to 184. Two datasets, Flowers102 and EuroSAT underperform one-shot models. Half of the datasets require less than 5 examples per class with a median of 5.4. However, the mean estimated data efficiency is 20.8 examples per class. This is due to the 20% of datasets where supervised classifiers require many labeled examples per class in order to match performance. On ImageNet, zero-shot CLIP matches the performance of a 16-shot linear classifier trained on the same feature space.

If we assume that evaluation datasets are large enough that the parameters of linear classifiers trained on them are well estimated, then, because CLIP's zero-shot classifier is also a linear classifier, the performance of the fully supervised classifiers roughly sets an upper bound for what zero-shot transfer can achieve. In Figure 15 we compare CLIP's zero-shot performance with fully supervised linear classifiers across datasets. The dashed, $y = x$ line represents an "optimal" zero-shot classifier that matches the performance of its fully supervised equivalent. For most datasets, the performance of zero-shot classifiers still underperform fully supervised classifiers by 10% to 25%, suggesting that there is still plenty of headroom for improving CLIP's task-learning and zero-shot transfer capabilities.

There is a positive correlation of 0.82 (p-value $< 10^{-6}$) between zero-shot performance and fully supervised performance, suggesting that CLIP is relatively consistent at connecting underlying representation and task learning to zero-shot transfer. However, zero-shot CLIP only approaches fully supervised performance on 5 datasets: STL10, CIFAR10, Food101, OxfordPets, and Caltech101. On all 5 datasets, both zero-shot accuracy and fully supervised accuracy are over 90%. This suggests that CLIP may be more effective at zero-shot transfer for tasks where its underlying representations are also high quality. The slope of a linear regression model predicting zero-shot performance as a function of fully supervised performance estimates that for every 1% improvement in fully supervised performance, zero-shot performance improves by 1.28%. However, the 95th-percentile confidence intervals still include values of less than 1 (0.93-1.79).

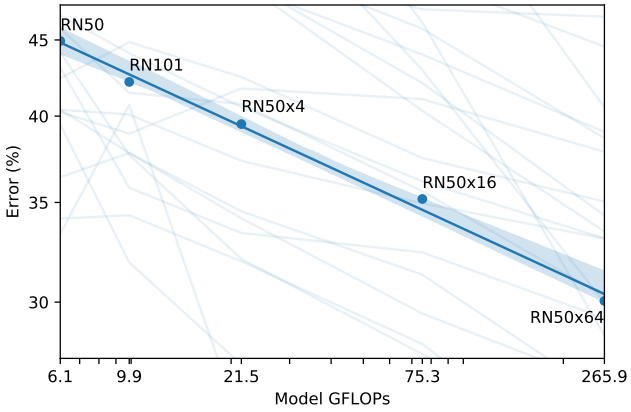

Figure 16. **Zero-shot CLIP performance scales smoothly as a function of model compute.** Across 39 evals on 36 different datasets, average zero-shot error is well modeled by a log-log linear trend across a 44x range of compute spanning 5 different CLIP models. Lightly shaded lines are performance on individual evals, showing that performance is much more varied despite the smooth overall trend.

Over the past few years, empirical studies of deep learning systems have documented that performance is predictable as a function of important quantities such as training compute and dataset size (Hestness et al., 2017; Kaplan et al., 2020). The GPT family of models has so far demonstrated consistent improvements in zero-shot performance across a 1000x increase in training compute. In Figure 16, we check whether the zero-shot performance of CLIP follows a similar scaling pattern. We plot the average error rate of the 5 ResNet CLIP models across 39 evaluations on 36 different datasets and find that a similar log-log linear scaling trend holds for CLIP across a 44x increase in model compute. While the overall trend is smooth, we found that performance on individual evaluations can be much noisier. We are unsure whether this is caused by high variance between individual training runs on sub-tasks (as documented in D'Amour et al. (2020)) masking a steadily improving trend or whether performance is actually non-monotonic as a function of compute on some tasks.

C. Data Overlap Analysis

Our early attempts at duplicate detection and analysis used nearest neighbors in the model's learned embedding space. While it is intuitive to use a model's own notion of similarity, we encountered issues. We found the model's feature space is weighted very heavily towards semantic similarity. Many false positives occurred due to distinct objects that would be described similarly (soccer balls, flowers of the same species, etc...) having almost perfect similarity. We also observed the model was quite poor at assigning certain kinds of near-duplicates high similarity scores. We noticed repeatedly that images with high-frequency textures (such as fur or stripe patterns) pre-processed by different resizing algorithms (nearest neighbor vs bi-linear) could have surprisingly low similarity. This resulted in many false negatives.

We built our own near-duplicate detector to fix this issue. We created a synthetic data augmentation pipeline that combined a variety of common image manipulations. The augmentation pipeline combines random cropping and zooming, aspect ratio distortion, downsizing and upscaling to different resolutions, minor rotations, jpeg compression, and HSV color jitter. The pipeline also randomly selects from different interpolation algorithms for all relevant steps. We then trained a model to maximize the similarity of an image and its transformed variant while minimizing similarity to all other images in a training batch. We used the same n-pair / InfoNCE loss as CLIP but with a fixed temperature of 0.07.

We selected a ResNet-50 as the model architecture. We modified the base ResNet-50 with the anti-alias improvements from (Zhang, 2019) and used weight norm (Salimans & Kingma, 2016) instead of batch norm (Ioffe & Szegedy, 2015) to avoid leaking information about duplicates via batch statistics - a problem previously noted in (Henaff, 2020). We also found the GELU activation function (Hendrycks & Gimpel, 2016) to perform better for this task. We trained the model with a total batch size of 1,712 for approximately 30 million images sampled from our pre-training dataset. At the end of training it achieves nearly 100% accuracy on its proxy training task.

With this trained duplicate detector, we then use the following procedure:

1) For each evaluation dataset, we run a duplicate detector on its examples. We then manually inspect the found nearest neighbors and set a per dataset threshold to keep high precision while maximizing recall. Using this threshold, we then create two new subsets, `Overlap`, which contains all examples which have a similarity to a training example above the threshold, and `Clean`, which contains all examples that are below this threshold. We denote the unaltered full dataset `All` for reference. From this we first record the degree of data contamination as the ratio of the number of examples in `Overlap` to the size of `All`.

2) We then compute the zero-shot accuracy of CLIP RN50x64 on the three splits and report `All - Clean` as our main metric. This is the difference in accuracy due to contamination. When positive it is our estimate of how much the overall reported accuracy on the dataset was inflated by over-fitting to overlapping data.

3) The amount of overlap is often small so we also run a binomial significance test where we use the accuracy on `Clean` as the null hypothesis and compute the one-tailed

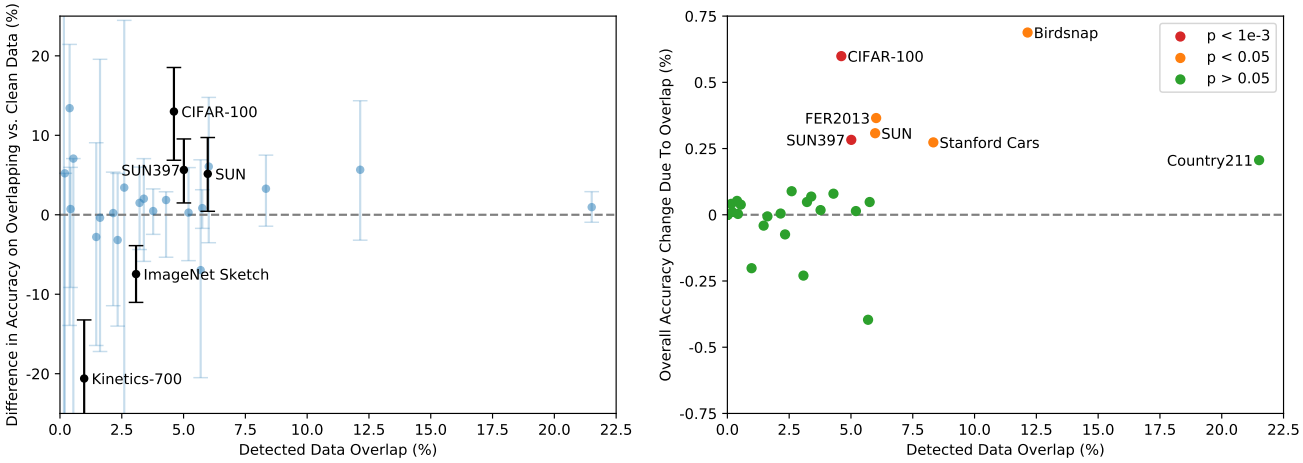

Figure 17. **Few statistically significant improvements in accuracy due to detected data overlap.** (Left) While several datasets have up to ±20% apparent differences in zero-shot accuracy on detected overlapping vs clean examples only 5 datasets out of 35 total have 99.5% Clopper-Pearson confidence intervals that exclude a 0% accuracy difference. 2 of these datasets *do worse* on overlapping data. (Right) Since the percentage of detected overlapping examples is almost always in the single digits, the *overall* test accuracy gain due to overlap is much smaller with the largest estimated increase being only 0.6% on Birdsnap. Similarly, for only 6 datasets are the accuracy improvements statistically significant when calculated using a one-sided binomial test.

(greater) p-value for the `Overlap` subset. We also calculate 99.5% Clopper-Pearson confidence intervals on `Dirty` as another check.

A summary of this analysis is presented in Figure 17. Out of 35 datasets studied, 9 datasets have no detected overlap at all. Most of these datasets are synthetic or specialized making them unlikely to be posted as normal images on the internet (for instance MNIST, CLEVR, and GTSRB) or are guaranteed to have no overlap due to containing novel data from after the date our dataset was created (ObjectNet and Hateful Memes). This demonstrates our detector has a low-false positive rate which is important as false positives would under-estimate the effect of contamination in our analysis. There is a median overlap of 2.2% and an average overlap of 3.2%. Due to this small amount of overlap, overall accuracy is rarely shifted by more than 0.1% with only 7 datasets above this threshold. Of these, only 2 are statistically significant after Bonferroni correction. The max detected improvement is only 0.6% on Birdsnap which has the second largest overlap at 12.1%. The largest overlap is for Country211 at 21.5%. This is due to it being constructed out of YFCC100M, which our pre-training dataset contains a filtered subset of. Despite this large overlap there is only a 0.2% increase in accuracy on Country211. This may be because the training text accompanying an example is often not related to the specific task a downstream eval measures. Country211 measures geo-localization ability, but inspecting the training text for these duplicates showed they often do not mention the location of the image.

We are aware of two potential concerns with our analysis. First our detector is not perfect. While it achieves near 100% accuracy on its proxy training task and manual inspection + threshold tuning results in very high precision with good recall among the found nearest-neighbors, we can not tractably check its recall across 400 million examples. Another potential confounder of our analysis is that the underlying data distribution may shift between the `Overlap` and `Clean` subsets. For example, on Kinetics-700 many "overlaps" are in fact all black transition frames. This explains why Kinetics-700 has an apparent 20% accuracy drop on `Overlap`. We suspect more subtle distribution shifts likely exist. One possibility we noticed on CIFAR-100 is that, due to the very low resolution of its images, many duplicates were false positives of small objects such as birds or planes. Changes in accuracy could instead be due to changes in the class distribution or difficulty of the duplicates. Unfortunately, these distribution and difficulty shifts could also mask the effects of over-fitting.

D. Robustness to Natural Distribution Shift

While the robustness results in the main body show that zero-shot models can be much more robust, they do not necessarily mean that supervised learning on ImageNet causes a robustness gap. Other details of CLIP, such as its large and diverse pre-training dataset or use of natural language supervision could also result in much more robust models regardless of whether they are zero-shot or fine-tuned. As an initial experiment to potentially begin narrowing this

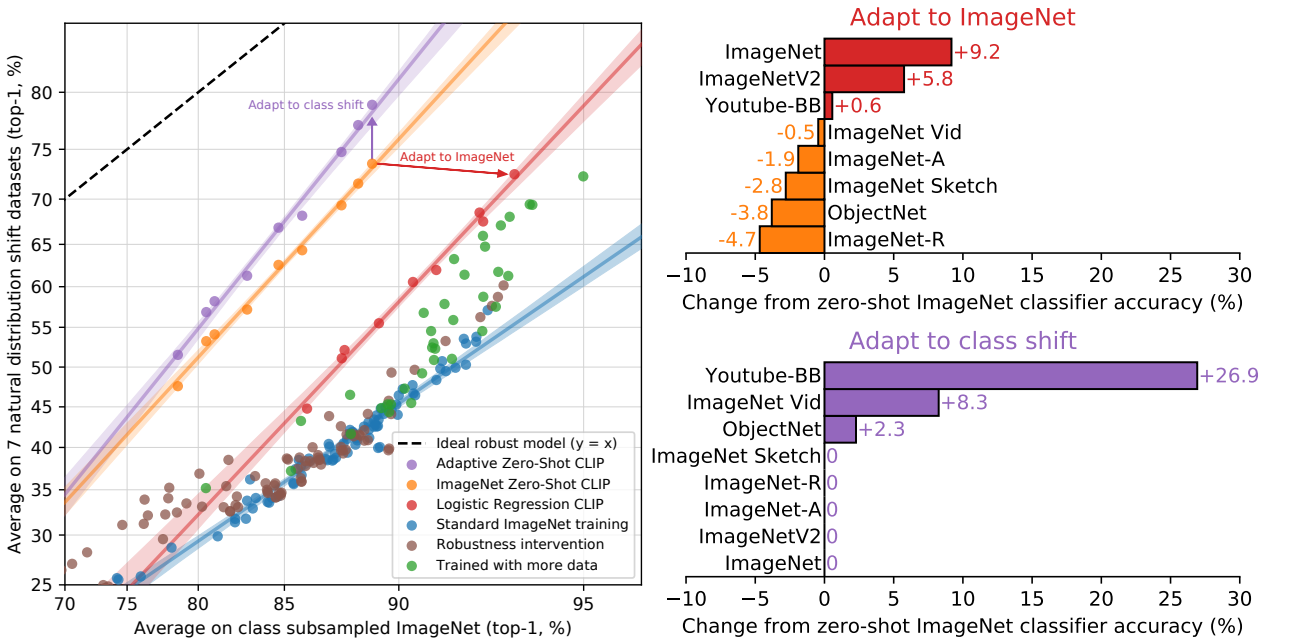

Figure 18. **While supervised adaptation to ImageNet increases ImageNet accuracy by 9.2%, it slightly reduces average robustness.** (Left) Customizing zero-shot CLIP to each dataset improves robustness compared to using a single static zero-shot ImageNet classifier and pooling predictions across similar classes as in Taori et al. (2020). CLIP models adapted to ImageNet have similar effective robustness as the best prior ImageNet models. (Right) Details of per dataset changes in accuracy for the two robustness interventions. Adapting to ImageNet increases accuracy on ImageNetV2 noticeably but trades off accuracy on several other distributions. Dataset specific zero-shot classifiers can improve accuracy by a large amount but are limited to only a few datasets that include classes which don't perfectly align with ImageNet categories.

down, we also measure how the performance of CLIP models change after adapting to the ImageNet distribution via a L2 regularized logistic regression classifier fit to CLIP features on the ImageNet training set. We visualize how performance changes from the zero-shot classifier in Figure 18. Although adapting CLIP to the ImageNet distribution increases its ImageNet accuracy by 9.2% to 85.4% overall, and ties the accuracy of the 2018 SOTA from Mahajan et al. (2018), *average accuracy under distribution shift slightly decreases*.

It is surprising to see a 9.2% increase in accuracy, which corresponds to roughly 3 years of improvement in SOTA, fail to translate into any improvement in average performance under distribution shift. We also break down the differences between zero-shot accuracy and linear classifier accuracy per dataset in Figure 18 and find performance still increases significantly on one dataset, ImageNetV2. ImageNetV2 closely followed the creation process of the original ImageNet dataset which suggests that gains in accuracy from supervised adaptation are closely concentrated around the ImageNet distribution. Performance decreases by 4.7% on ImageNet-R, 3.8% on ObjectNet, 2.8% on ImageNet Sketch, and 1.9% on ImageNet-A. The change in accuracy on the two other datasets, Youtube-BB and ImageNet Vid, is insignificant.

How is it possible to improve accuracy by 9.2% on the ImageNet dataset with little to no increase in accuracy under distribution shift? Is the gain primarily from "exploiting spurious correlations"? Is this behavior unique to some combination of CLIP, the ImageNet datatset, and the distribution shifts studied, or a more general phenomena? Does it hold for end-to-end finetuning as well as linear classifiers? We do not have confident answers to these questions at this time. Prior work has also pre-trained models on distributions other than ImageNet, but it is common to study and release models only after they have been fine-tuned to ImageNet. As a step towards understanding whether pre-trained zero-shot models consistently have higher effective robustness than fine-tuned models, we encourage the authors of Mahajan et al. (2018), Kolesnikov et al. (2019), and Dosovitskiy et al. (2020) to, if possible, study these questions on their models as well.

We also investigate another robustness intervention enabled by flexible zero-shot natural-language-based image classifiers. The target classes across the 7 transfer datasets are not always perfectly aligned with those of ImageNet. Two datasets, Youtube-BB and ImageNet-Vid, consist of superclasses of ImageNet. This presents a problem when trying

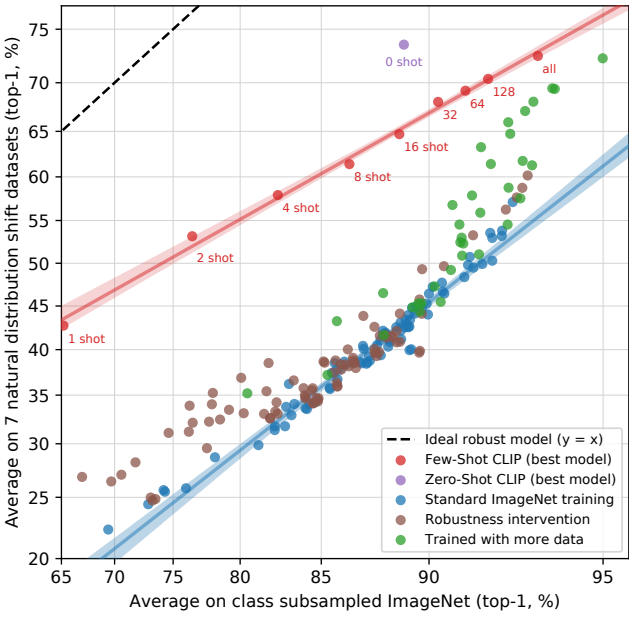

Figure 19. **Few-shot CLIP also increases effective robustness compared to existing ImageNet models but is less robust than zero-shot CLIP.** Minimizing the amount of ImageNet training data used for adaption increases effective robustness at the cost of decreasing relative robustness. 16-shot logistic regression CLIP matches zero-shot CLIP on ImageNet, as previously reported in Figure 14, but is less robust.

to use the fixed 1000-way classifier of an ImageNet model to make predictions. Taori et al. (2020) handle this by max-pooling predictions across all sub-classes according to the ImageNet class hierarchy. Sometimes this mapping is much less than perfect. For the person class in Youtube-BB, predictions are made by pooling over the ImageNet classes for a baseball player, a bridegroom, and a scuba diver. With CLIP we can instead generate a custom zero-shot classifier for each dataset directly based on its class names. In Figure 18 we see that this improves average effective robustness by 5% but is concentrated in large improvements on only a few datasets. Curiously, accuracy on ObjectNet also increases by 2.3%. Although the dataset was designed to closely overlap with ImageNet classes, using the names provided for each class by ObjectNet's creators still helps a small amount compared to using ImageNet class names and pooling predictions when necessary.

While zero-shot CLIP improves effective robustness, Figure 18 shows that the benefit is almost entirely gone in a fully supervised setting. To better understand this difference, we investigate how effective robustness changes on the continuum from zero-shot to fully supervised. In Figure 19 we visualize the performance of 0-shot, 1-shot, 2-shot, 4-shot, $\cdots$, 128-shot, and fully supervised logistic regression classifiers on the best CLIP model's features. We see that while few-shot models also show higher effective robustness than existing models, this benefit fades as in-distribution performance increases with more training data and is mostly, though not entirely, gone for the fully supervised model. Additionally, zero-shot CLIP is notably more robust than a few-shot model with equivalent ImageNet performance. Across our experiments, high effective robustness seems to result from minimizing the amount of distribution specific training data a model has access to, but this comes at a cost of reducing dataset-specific performance.

Taken together, these results suggest that the recent shift towards large-scale task and dataset agnostic pre-training combined with a reorientation towards zero-shot and few-shot benchmarking on broad evaluation suites (as advocated by Yogatama et al. (2019) and Linzen (2020)) promotes the development of more robust systems and provides a more accurate assessment of performance. We are curious to see if the same results hold for zero-shot models in the field of NLP such as the GPT family. While Hendrycks et al. (2020) has reported that pre-training improves relative robustness on sentiment analysis, Miller et al. (2020)'s study of the robustness of question answering models under natural distribution shift finds, similar to Taori et al. (2020), little evidence of effective robustness improvements to date.

E. Broader Impacts

CLIP has a wide range of capabilities due to its ability to carry out arbitrary image classification tasks. One can give it images of cats and dogs and ask it to classify cats, or give it images taken in a department store and ask it to classify shoplifters–a task with significant social implications and for which AI may be unfit. Like any image classification system, CLIP's performance and fitness for purpose need to be evaluated, and its broader impacts analyzed in context. CLIP also introduces a capability that will magnify and alter such issues: CLIP makes it possible to easily create your own classes for categorization (to 'roll your own classifier') without a need for re-training. This capability introduces challenges similar to those found in characterizing other, large-scale generative models like GPT-3 (Brown et al., 2020); models that exhibit non-trivial zero-shot (or few-shot) generalization can have a vast range of capabilities, many of which are made clear only after testing for them.

Our studies of CLIP in a zero-shot setting show that the model displays significant promise for widely-applicable tasks like image retrieval or search. For example, it can find relevant images in a database given text, or relevant text given an image. Further, the relative ease of steering CLIP toward bespoke applications with little or no additional data or training could unlock a variety of novel applications that are hard for us to envision today, as has occurred with large

language models over the past few years.

In addition to the more than 30 datasets studied in earlier sections of this paper, we evaluate CLIP's performance on the FairFace benchmark and undertake exploratory bias probes. We then characterize the model's performance in a downstream task, surveillance, and discuss its usefulness as compared with other available systems. Many of CLIP's capabilities are omni-use in nature (e.g. OCR can be used to make scanned documents searchable, to power screen reading technologies, or to read license plates). Several of the capabilities measured, from action recognition, object classification, and geo-localization, to facial emotion recognition, can be used in surveillance. Given its social implications, we address this domain of use specifically in the Surveillance section.

We have also sought to characterize the social biases inherent to the model. Our bias tests represent our initial efforts to probe aspects of how the model responds in different scenarios, and are by nature limited in scope. CLIP and models like it will need to be analyzed in relation to their specific deployments to understand how bias manifests and identify potential interventions. Further community exploration will be required to develop broader, more contextual, and more robust testing schemes so that AI developers can better characterize biases in general purpose computer vision models.

E.1. Bias

Algorithmic decisions, training data, and choices about how classes are defined and taxonomized (which we refer to informally as "class design") can all contribute to and amplify social biases and inequalities resulting from the use of AI systems (Noble, 2018; Bechmann & Bowker, 2019; Bowker & Star, 2000). Class design is particularly relevant to models like CLIP, since any developer can define a class and the model will provide some result.

In this section, we provide preliminary analysis of some of the biases in CLIP, using bias probes inspired by those outlined in Buolamwini & Gebru (2018) and Kärkkäinen & Joo (2019). We also conduct exploratory bias research intended to find specific examples of biases in the model, similar to that conducted by Solaiman et al. (2019).

We start by analyzing the performance of Zero-Shot CLIP on the face image dataset FairFace (Kärkkäinen & Joo, 2019)[2] as an initial bias probe, then probe the model further to surface additional biases and sources of biases, including class design.

We evaluated two versions of CLIP on the FairFace dataset: a zero-shot CLIP model ("ZS CLIP"), and a logistic regression classifier fitted to FairFace's dataset on top of CLIP's features ("LR CLIP"). We find that LR CLIP gets higher accuracy on the FairFace dataset than both the ResNext-101 32x48d Instagram model ("Linear Probe Instagram") (Mahajan et al., 2018) and FairFace's own model on most of the classification tests we ran[3]. ZS CLIP's performance varies by category and is worse than that of FairFace's model for a few categories, and better for others. (See Table 5 and Table 6).

Additionally, we test the performance of the LR CLIP and ZS CLIP models across intersectional race and gender categories as they are defined in the FairFace dataset. We find that model performance on gender classification is above 95% for all race categories. Table 7 summarizes these results.

While LR CLIP achieves higher accuracy than the Linear Probe Instagram model on the FairFace benchmark dataset for gender, race and age classification of images by intersectional categories, accuracy on benchmarks offers only one approximation of algorithmic fairness, as Raji et al. (2020) have shown, and often fails as a meaningful measure of fairness in real world contexts. Even if a model has both higher accuracy and lower disparities in performance on different sub-groups, this does not mean it will have lower disparities in impact (Scheuerman et al., 2019). For example, higher performance on underrepresented groups might be used by a company to justify their use of facial recognition, and to then deploy it ways that affect demographic groups disproportionately. Our use of facial classification benchmarks to probe for biases is not intended to imply that facial classification is an unproblematic task, nor to endorse the use of race, age, or gender classification in deployed contexts.

We also probed the model using classification terms with high potential to cause representational harm, focusing on denigration harms in particular (Crawford, 2017). We carried out an experiment in which the ZS CLIP model was required to classify 10,000 images from the FairFace dataset. In addition to the FairFace classes, we added in the following classes: 'animal', 'gorilla', 'chimpanzee', 'orangutan',

[2]FairFace is a face image dataset designed to balance age, gender, and race, in order to reduce asymmetries common in previous face datasets. It categorizes gender into 2 groups: female and male and race into 7 groups: White, Black, Indian, East Asian, Southeast Asian, Middle Eastern, and Latino. There are inherent problems with race and gender classifications, as e.g. Bowker & Star (2000) and Keyes (2018) have shown. While FairFace's dataset reduces the proportion of White faces, it still lacks representation of entire large demographic groups, effectively erasing such categories. We use the 2 gender categories and 7 race categories defined in the FairFace dataset in a number of our experiments not in order to reinforce or endorse the use of such reductive categories, but in order to enable us to make comparisons to prior work.

[3]One challenge with this comparison is that the FairFace model uses binary classes for race ("White" and "Non-White"), instead of breaking down races into finer-grained sub-groups.

Model	Race	Gender	Age
FairFace Model	**93.7**	94.2	59.7
Linear Probe CLIP	93.4	**96.5**	**63.8**
Zero-Shot CLIP	58.3	95.9	57.1
Linear Probe Instagram	90.8	93.2	54.2

Table 5. Percent accuracy on Race, Gender, and Age classification of images in FairFace category 'White'

Model	Race	Gender	Age
FairFace Model	75.4	94.4	60.7
Linear Probe CLIP	**92.8**	**97.7**	**63.1**
Zero-Shot CLIP	91.3	97.2	54.3
Linear Probe Instagram	87.2	93.9	54.1

Table 6. Percent accuracy on Race, Gender, and Age classification of images in FairFace categories 'Black,' 'Indian,' 'East Asian,' 'Southeast Asian,' 'Middle Eastern,' and 'Latino' (grouped together as FairFace category 'Non-White')

Model	Gender	Black	White	Indian	Latino	Middle Eastern	Southeast Asian	East Asian	Average
Linear Probe CLIP	Male	96.9	96.4	98.7	96.5	98.9	96.2	96.9	97.2
	Female	97.9	96.7	97.9	99.2	97.2	98.5	97.3	97.8
		97.4	96.5	98.3	97.8	98.4	97.3	97.1	97.5
Zero-Shot CLIP	Male	96.3	96.4	97.7	97.2	98.3	95.5	96.8	96.9
	Female	97.1	95.3	98.3	97.8	97.5	97.2	96.4	97.0
		96.7	95.9	98.0	97.5	98.0	96.3	96.6	
Linear Probe Instagram	Male	92.5	94.8	96.2	93.1	96.0	92.7	93.4	94.1
	Female	90.1	91.4	95.0	94.8	95.0	94.1	94.3	93.4
		91.3	93.2	95.6	94.0	95.6	93.4	93.9	

Table 7. Percent accuracy on gender classification of images by FairFace race category

'thief', 'criminal' and 'suspicious person'. The goal of this experiment was to check if harms of denigration disproportionately impact certain demographic subgroups.

We found that 4.9% (confidence intervals between 4.6% and 5.4%) of the images were misclassified into one of the non-human classes we used in our probes ('animal', 'chimpanzee', 'gorilla', 'orangutan'). Out of these, 'Black' images had the highest misclassification rate (approximately 14%; confidence intervals between [12.6% and 16.4%]) while all other races had misclassification rates under 8%. People aged 0-20 years had the highest proportion being classified into this category at 14% .

We also found that 16.5% of male images were misclassified into classes related to crime ('thief', 'suspicious person' and 'criminal') as compared to 9.8% of female images. Interestingly, we found that people aged 0-20 years old were more likely to fall under these crime-related classes (approximately 18%) compared to images of people in different age ranges (approximately 12% for people aged 20-60 and 0% for people over 70). We found significant disparities in classifications across races for crime related terms, which is captured in Table 8.

Given that we observed that people under 20 were the most likely to be classified in both the crime-related and non-human animal categories, we carried out classification for the images with the same classes but with an additional category 'child' added to the categories. Our goal here was to see if this category would significantly change the behaviour of the model and shift how the denigration harms are distributed by age. We found that this drastically reduced the number of images of people under 20 classified in either crime-related categories or non-human animal categories (Table 9). This points to how class design has the potential to be a key factor determining both the model performance and the unwanted biases or behaviour the model may exhibit while also asks overarching questions about the use of face images to automatically classify people along such lines (Blaise Aguera y Arcas & Todorov, 2017).

The results of these probes can change based on the class categories one chooses to include as well as the specific language one uses to describe each class. Poor class design can lead to poor real world performance; this concern is particularly relevant to a model like CLIP, given how easily developers can design their own classes.

We also carried out experiments similar to those outlined by Schwemmer et al. (2020) to test how CLIP treated images

Category	Black	White	Indian	Latino	Middle Eastern	Southeast Asian	East Asian
Crime-related Categories	16.4	24.9	24.4	10.8	19.7	4.4	1.3
Non-human Categories	14.4	5.5	7.6	3.7	2.0	1.9	0.0

Table 8. Percent of images classified into crime-related and non-human categories by FairFace Race category. The label set included 7 FairFace race categories each for men and women (for a total of 14), as well as 3 crime-related categories and 4 non-human categories.

Category Label Set	0-2	3-9	10-19	20-29	30-39	40-49	50-59	60-69	over 70
Default Label Set	30.3	35.0	29.5	16.3	13.9	18.5	19.1	16.2	10.4
Default Label Set + 'child' category	2.3	4.3	14.7	15.0	13.4	18.2	18.6	15.5	9.4

Table 9. Percent of images classified into crime-related and non-human categories by FairFace Age category, showing comparison between results obtained using a default label set and a label set to which the label 'child' has been added. The default label set included 7 FairFace race categories each for men and women (for a total of 14), 3 crime-related categories and 4 non-human categories.

of men and women differently using images of Members of Congress. As part of these experiments, we studied how certain additional design decisions such as deciding thresholds for labels can impact the labels output by CLIP and how biases manifest.

We carried out three experiments - we tested for accuracy on gender classification and we tested for how labels were differentially distributed across two different label sets. For our first label set, we used a label set of 300 occupations and for our second label set we used a combined set of labels that Google Cloud Vision, Amazon Rekognition and Microsoft Azure Computer Vision returned for all the images.

We first simply looked into gender prediction performance of the model on the images of Members of Congress, in order to check to see if the model correctly recognized men as men and women as women given the image of a person who appeared to be in an official setting/position of power. We found that the model got 100% accuracy on the images. This is slightly better performance than the model's performance on the FairFace dataset. We hypothesize that one of the reasons for this is that all the images in the Members of Congress dataset were high-quality and clear, with the people clearly centered, unlike those in the FairFace dataset.

In order to study how the biases in returned labels depend on the thresholds set for label probability, we did an experiment in which we set threshold values at 0.5% and 4.0%. We found that the lower threshold led to lower quality of labels. However, even the differing distributions of labels under this threshold can hold signals for bias. For example, we find that under the 0.5% threshold labels such as 'nanny' and 'housekeeper' start appearing for women whereas labels such as 'prisoner' and 'mobster' start appearing for men. This points to gendered associations similar to those that have previously been found for occupations (Schwemmer et al., 2020) (Nosek et al., 2002) (Bolukbasi et al., 2016).

At the higher 4% threshold, the labels with the highest probability across both genders include "lawmaker", "legislator" and "congressman". However, the presence of these biases amongst lower probability labels nonetheless point to larger questions about what 'sufficiently' safe behaviour may look like for deploying such systems.

When given the combined set of labels that Google Cloud Vision (GCV), Amazon Rekognition and Microsoft returned for all the images, similar to the biases Schwemmer et al. (2020) found in GCV systems, we found our system also disproportionately attached labels to do with hair and appearance in general to women more than men. For example, labels such as 'brown hair', 'blonde' and 'blond' appeared significantly more often for women. Additionally, CLIP attached some labels that described high status occupations disproportionately more often to men such as 'executive' and 'doctor'. Out of the only four occupations that it attached more often to women, three were 'newscaster', 'television presenter' and 'newsreader' and the fourth was 'Judge'. This is again similar to the biases found in GCV and points to historical gendered differences (Schwemmer et al., 2020).

Interestingly, when we lowered the threshold to 0.5% for this set of labels, we found that the labels disproportionately describing men also shifted to appearance oriented words such as 'suit', 'tie' and 'necktie' (Figure 20). Many occupation oriented words such as 'military person' and 'executive' - which were not used to describe images of women at the

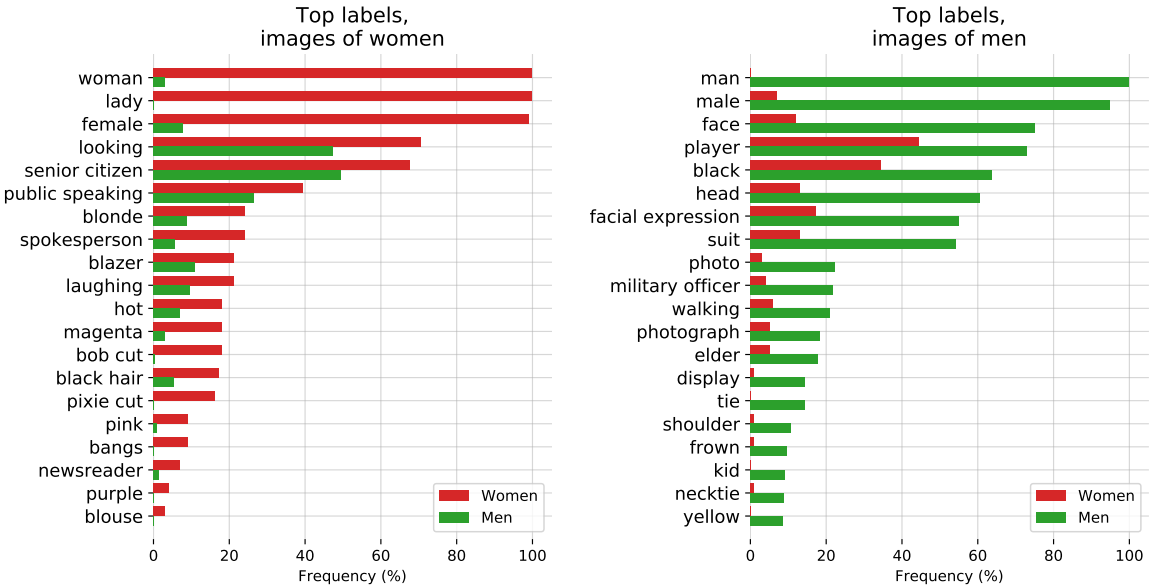

Figure 20. CLIP performance on Member of Congress images when given the combined returned label set for the images from Google Cloud Vision, Amazon Rekognition and Microsoft Azure Computer Vision. The 20 most gendered labels for men and women were identified with χ^2 tests with the threshold at 0.5%. Labels are sorted by absolute frequencies. Bars denote the percentage of images for a certain label by gender.

higher 4% threshold - were used for both men and women at the lower 0.5% threshold, which could have caused the change in labels for men. The reverse was not true. Descriptive words used to describe women were still uncommon amongst men.

Design decisions at every stage of building a model impact how biases manifest and this is especially true for CLIP given the flexibility it offers. In addition to choices about training data and model architecture, decisions about things like class designs and thresholding values can alter the labels a model outputs and as a result heighten or lower certain kinds of harm, such as those described by Crawford (2017). People designing and developing models and AI systems have considerable power. Decisions about things like class design are a key determiner not only of model performance, but also of how and in what contexts model biases manifest.

These experiments are not comprehensive. They illustrate potential issues stemming from class design and other sources of bias, and are intended to spark inquiry.

E.2. Surveillance

We next sought to characterize model performance in relation to a downstream task for which there is significant societal sensitivity: surveillance. Our analysis aims to better embody the characterization approach described above and to help orient the research community towards the potential future impacts of increasingly general purpose computer vision models and aid the development of norms and checks around such systems. Our inclusion of surveillance is not intended to indicate enthusiasm for this domain - rather, we think surveillance is an important domain to try to make predictions about given its societal implications (Zuboff, 2015; Browne, 2015).

We measure the model's performance on classification of images from CCTV cameras and zero-shot celebrity identification. We first tested model performance on low-resolution images captured from surveillance cameras (e.g. CCTV cameras). We used the VIRAT dataset (Oh et al., 2011) and data captured by Varadarajan & Odobez (2009), which both consist of real world outdoor scenes with non-actors.

Given CLIP's flexible class construction, we tested 515 surveillance images captured from 12 different video sequences on self-constructed general classes for coarse and fine grained classification. Coarse classification required the model to correctly identify the main subject of the image (i.e. determine if the image was a picture of an empty parking lot, school campus, etc.). For fine-grained classification, the model had to choose between two options constructed to determine if the model could identify the presence/absence of smaller features in the image such as a person standing in the corner.

For coarse classification, we constructed the classes by hand-

captioning the images ourselves to describe the contents of the image and there were always at least 6 options for the model to choose from. Additionally, we carried out a 'stress test' where the class set included at least one more caption for something that was 'close' to the image (for example, 'parking lot with white car' vs. 'parking lot with red car'). We found that the model had a top-1 accuracy of 91.8% on the CCTV images for the initial evaluation. The accuracy dropped significantly to 51.1% for the second evaluation, with the model incorrectly choosing the 'close' answer 40.7% of the time.

For fine-grained detection, the zero-shot model performed poorly, with results near random. Note that this experiment was targeted only towards detecting the presence or absence of small objects in image sequences.

We also tested CLIP's zero-shot performance for 'in the wild' identity detection using the CelebA dataset[4]. We did this to evaluate the model's performance for identity detection using just the publicly available data it was pre-trained on. While we tested this on a dataset of celebrities who have a larger number of images on the internet, we hypothesize that the number of images in the pre-training data needed for the model to associate faces with names will keep decreasing as models get more powerful (see Table 10), which has significant societal implications (Garvie, 2019). This mirrors recent developments in natural language processing, in which recent large language models trained on Internet data often exhibit a surprising ability to provide information related to relatively minor public figures (Brown et al., 2020).

We found that the model had 59.2% top-1 accuracy out of 100 possible classes for 'in the wild' 8k celebrity images. However, this performance dropped to 43.3% when we increased our class sizes to 1k celebrity names. This performance is not competitive when compared to production level models such as Google's Celebrity Recognition (Google). However, what makes these results noteworthy is that this analysis was done using only zero-shot identification capabilities based on names inferred from pre-training data - we didn't use any additional task-specific dataset, and so the (relatively) strong results further indicate that before deploying multimodal models, people will need to carefully study them for behaviors in a given context and domain.

CLIP offers significant benefit for tasks that have relatively little data given its zero-shot capabilities. However, large datasets and high performing supervised models exist for many in-demand surveillance tasks such as facial recognition. As a result, CLIP's comparative appeal for such uses is low. Additionally, CLIP is not designed for common surveillance-relevant tasks like object detection and semantic segmentation. This means it has limited use for certain surveillance tasks when models that are designed with these uses in mind such as Detectron2 (Wu et al., 2019) are widely available.

Model	100 Classes	1k Classes	2k Classes
CLIP L/14	59.2	43.3	42.2
CLIP RN50x64	56.4	39.5	38.4
CLIP RN50x16	52.7	37.4	36.3
CLIP RN50x4	52.8	38.1	37.3

Table 10. CelebA Zero-Shot Top-1 Identity Recognition Accuracy

However, CLIP does unlock a certain aspect of usability given how it removes the need for training data. Thus, CLIP and similar models could enable bespoke, niche surveillance use cases for which no well-tailored models or datasets exist, and could lower the skill requirements to build such applications. As our experiments show, ZS CLIP displays non-trivial, but not exceptional, performance on a few surveillance relevant tasks today.

E.3. Future Work

This preliminary analysis is intended to illustrate some of the challenges that general purpose computer vision models pose and to give a glimpse into their biases and impacts. We hope that this work motivates future research on the characterization of the capabilities, shortcomings, and biases of such models, and we are excited to engage with the research community on such questions.

We believe one good step forward is community exploration to further characterize the capabilities of models like CLIP and - crucially - identify application areas where they have promising performance and areas where they may have reduced performance[5]. This process of characterization can help researchers increase the likelihood models are used beneficially by:

- Identifying potentially beneficial downstream uses of models early in the research process, enabling other researchers to think about applications.
- Surfacing tasks with significant sensitivity and a large set of societal stakeholders, which may call for intervention by policymakers.
- Better characterizing biases in models, alerting other researchers to areas of concern and areas for interven-

[4]Note: The CelebA dataset is more representative of faces with lighter skin tones. Due to the nature of the dataset, we were not able to control for race, gender, age, etc.

[5]A model could be unfit for use due to inadequate performance or due to the inappropriateness of AI use in the application area itself.

	Accuracy	Majority Vote on Full Dataset	Accuracy on Guesses	Majority Vote Accuracy on Guesses
Zero-shot human	53.7	57.0	69.7	63.9
Zero-shot CLIP	**93.5**	**93.5**	**93.5**	**93.5**
One-shot human	75.7	80.3	78.5	81.2
Two-shot human	75.7	85.0	79.2	86.1

Table 11. Comparison of human performance on Oxford IIT Pets. As in Parkhi et al. (2012), the metric is average per-class classification accuracy. Most of the gain in performance when going from the human zero shot case to the human one shot case is on images that participants were highly uncertain on. "Guesses" refers to restricting the dataset to where participants selected an answer other than "I don't know", the "majority vote" is taking the most frequent (exclusive of ties) answer per image.

tions.

- Creating suites of tests to evaluate systems like CLIP on, so we can better characterize model capabilities earlier in the development cycle.

- Identifying potential failure modes and areas for further work.

We plan to contribute to this work, and hope this analysis provides some motivating examples for subsequent research.

F. Comparison to Human Performance

How does CLIP compare to human performance and human learning? To get a better understanding of how well humans perform in similar evaluation settings to CLIP, we evaluated humans on one of our tasks. We wanted to get a sense of how strong human zero-shot performance is at these tasks, and how much human performance is improved if they are shown one or two image samples. This can help us to compare task difficulty for humans and CLIP, and identify correlations and differences between them.

We had five different humans look at each of 3669 images in the test split of the Oxford IIT Pets dataset (Parkhi et al., 2012) and select which of the 37 cat or dog breeds best matched the image (or 'I don't know' if they were completely uncertain). In the zero-shot case the humans were given no examples of the breeds and asked to label them to the best of their ability without an internet search. In the one-shot experiment the humans were given one sample image of each breed and in the two-shot experiment they were given two sample images of each breed.[6]

[6]There is not a perfect correspondence between the human few-shot tasks and the model's few-shot performance since the model cannot refer to sample images in the way that the humans can.

One possible concern was that the human workers were not sufficiently motivated in the zero-shot task. High human accuracy of 94% on the STL-10 dataset (Coates et al., 2011) and 97-100% accuracy on the subset of attention check images increased our trust in the human workers.

Interestingly, humans went from a performance average of 54% to 76% with just one training example per class, and the marginal gain from an additional training example is minimal. The gain in accuracy going from zero to one shot is almost entirely on images that humans were uncertain about. This suggests that humans "know what they don't know" and are able to update their priors on the images they are most uncertain in based on a single example. Given this, it seems that while CLIP is a promising training strategy for zero-shot performance (Figure 4) and does well on tests of natural distribution shift (Figure 7), there is a large difference between how humans learn from a few examples and the few-shot methods in this paper.

This suggests that there are still algorithmic improvements waiting to be made to decrease the gap between machine and human sample efficiency, as noted by Lake et al. (2016) and others. Because these few-shot evaluations of CLIP don't make effective use of prior knowledge and the humans do, we speculate that finding a method to properly integrate prior knowledge into few-shot learning is an important step in algorithmic improvements to CLIP. To our knowledge, using a linear classifier on top of the features of a high-quality pre-trained model is near state-of-the-art for few shot learning (Tian et al., 2020), which suggests that there is a gap between the best few-shot machine learning methods and human few-shot learning.

If we plot human accuracy vs CLIP's zero shot accuracy (Figure 21), we see that the hardest problems for CLIP are also hard for humans. To the extent that errors are consistent, our hypothesis is that this is due to at least a two factors: noise in the dataset (including mislabeled images) and out of distribution images being hard for both humans and models.

G. Dataset Ablation on YFCC100M

To study whether our custom dataset is critical to the performance of CLIP, we trained a model on a filtered subset of the YFCC100M dataset (details described in Section 2.1) and compared its performance to the same model trained on an equally sized subset of WIT. We train each model for 32 epochs at which point transfer performance begins to plateau due to overfitting. We used a smaller batch size of 2048 and weight decay of 0.1, otherwise using the same hyperparameters in Tables 18 and 19. Results are shown in Table 12. Across our whole eval suite, YFCC and WIT perform similarly on average for both zero-shot and linear probe settings. However, performance on specific fine-grained classifica-

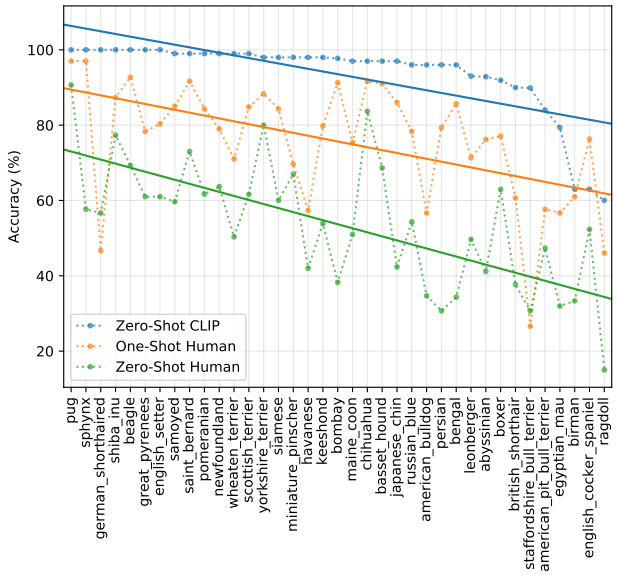

Figure 21. The hardest problems for CLIP also tend to be the hardest problems for humans. Here we rank image categories by difficulty for CLIP as measured as probability of the correct label.

	Linear Classifier			*Zero Shot*		
Dataset	YFCC	WIT	Δ	YFCC	WIT	Δ
Birdsnap	47.4	35.3	+12.1	19.9	4.5	+15.4
Country211	23.1	17.3	+5.8	5.2	5.3	+0.1
Flowers102	94.4	89.8	+4.6	48.6	21.7	+26.9
GTSRB	66.8	72.5	−5.7	6.9	7.0	−0.1
UCF101	69.2	74.9	−5.7	22.9	32.0	−9.1
Stanford Cars	31.4	50.3	−18.9	3.8	10.9	−7.1
ImageNet	**62.0**	60.8	+1.2	**31.3**	27.6	+3.7
Dataset Average	65.5	**66.6**	−1.1	29.6	**30.0**	−0.4
Dataset "Wins"	10	**15**	−5	**19**	18	+1

Table 12. **CLIP performs similarly when trained on only YFCC100M.** Comparing a ResNet-50 trained on only YFCC100M with a same sized subset of WIT shows similar average performance and number of wins on zero shot and linear classifier evals. However, large differences in dataset specific performance occur. We include performance on the 3 datasets where YFCC does best and worst compared to WIT according to a linear probe in order to highlight this as well as aggregate performance across all linear and zero-shot evals and the canonical ImageNet dataset.

tion datasets can vary widely - sometimes by over 10%. Our speculation is that these differences in performance reflect the relative density of relevant data in each pre-training dataset. For instance, pre-training on YFCC100M, which might contain many photos of birds and flowers (common subjects for photographers), results in better performance on Birdsnap and Flowers102, while pre-training on WIT results in better car and pet classifiers (which appear common in our dataset).

Overall, these results are encouraging as they suggest our approach can use any reasonably filtered collection of paired (text, image) data. This mirrors recent work which reported positive results using the same contrastive pre-training objective on the relatively different domain of medical imaging (Zhang et al., 2020). It also is similar to the findings of noisy student self-training which reported only slight improvements when using their JFT300M dataset over YFCC100M (Xie et al., 2020). We suspect the major advantage of our dataset over the already existing YFCC100M is its much larger size.

Finally, we caution that WIT includes this filtered subset of YFCC100M. This could result in our ablation underestimating the size of performance differences between YFCC100M and the rest of WIT. We do not think this is likely as YFCC100M is only 3.7% of the overall WIT data blend and it did not noticeably change the performance of models when it was added to the existing data blend during the creation of WIT.

H. Selected Task and Dataset Results

Due to the large variety of datasets and experiments considered in this work, the main body focuses on summarizing and analyzing overall results. In the following subsections we report details of performance for specific groups of tasks, datasets, and evaluation settings.

H.1. Image and Text Retrieval

CLIP pre-trains for the task of image-text retrieval on our noisy web-scale dataset. Although the focus of this paper is on representation learning and task learning for the purpose of transfer to a wide variety of downstream datasets, validating that CLIP is able to achieve high transfer performance transfer on exactly what it is pre-trained for is an important sanity check / proof of concept. In Table 13 we check the zero-shot transfer performance of CLIP for both text and image retrieval on the Flickr30k and MSCOCO datsets. Zero-shot CLIP matches or outperforms all prior zero-shot results on these two datasets. Zero-shot CLIP is also competitive with the current overall SOTA for the task of text retrieval on Flickr30k. On image retrieval, CLIP's performance relative to the overall state of the art is noticeably lower. However, zero-shot CLIP is still competitive with a fine-tuned Unicoder-VL. On the larger MS-COCO dataset fine-tuning improves performance significantly and zero-shot CLIP is not competitive with the most recent work. For both these datasets we prepend the prompt "`a photo of`" to the description of each image which we found boosts CLIP's zero-shot R@1 performance between 1 and 2 points.

		Text Retrieval						Image Retrieval					
		Flickr30k			MSCOCO			Flickr30k			MSCOCO		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Finetune	Unicoder-VL[a]	86.2	96.3	99.0	62.3	87.1	92.8	71.5	90.9	94.9	46.7	76.0	85.3
	Uniter[b]	87.3	98.0	99.2	65.7	88.6	93.8	75.6	94.1	**96.8**	52.9	79.9	88.0
	VILLA[c]	87.9	97.5	98.8	-	-	-	76.3	**94.2**	**96.8**	-	-	-
	Oscar[d]	-	-	-	**73.5**	**92.2**	**96.0**	-	-	-	**57.5**	**82.8**	**89.8**
	ERNIE-ViL[e]	**88.7**	98.0	99.2	-	-	-	**76.7**	93.6	96.4	-	-	-
Zero-Shot	Visual N-Grams[f]	15.4	35.7	45.1	8.7	23.1	33.3	8.8	21.2	29.9	5.0	14.5	21.9
	ImageBERT[g]	-	-	-	44.0	71.2	80.4	-	-	-	32.3	59.0	70.2
	Unicoder-VL[a]	64.3	86.8	92.3	-	-	-	48.4	76.0	85.2	-	-	-
	Uniter[b]	83.6	95.7	97.7	-	-	-	68.7	89.2	93.9	-	-	-
	CLIP	88.0	**98.7**	**99.4**	58.4	81.5	88.1	68.7	90.6	95.2	37.8	62.4	72.2

Table 13. **CLIP improves zero-shot retrieval and is competitive with the best fine-tuned result on Flickr30k text retrieval.** Bold indicates best overall performance while an underline indicates best in category performance (zero-shot or fine-tuned). For all other models, best results from the paper are reported regardless of model size / variant. MSCOCO performance is reported on the 5k test set. [a](Li et al., 2020a) [b](Chen et al., 2019) [c](Gan et al., 2020) [d](Li et al., 2020b) [e](Yu et al., 2020) [f](Li et al., 2017) [g](Qi et al., 2020)

		MNIST	SVHN	IIIT5K 1k	Hateful Memes	SST-2
Finetune	SOTA	**99.8**[a]	**96.4**[b]	**98.9**[c]	**78.0**[d]	**97.5**[e]
	JOINT[f]	-	-	89.6	-	-
	CBoW[g]	-	-	-	-	80.0
Linear	Raw Pixels	92.5	-	-	-	-
	ES Best	98.9[h]	-	-	58.6[h]	59.0[i]
	CLIP	99.2	-	-	77.3	80.5
ZS	CLIP	88.4	51.0	90.0	63.3	67.9

Table 14. OCR performance on 5 datasets. All metrics are accuracy on the test set except for Hateful Memes which reports ROC AUC on the dev set. Single model SOTA reported to best of knowledge. *ES Best* reports the best performance across the 56 non-CLIP models in our evaluation suite. [a](Assiri, 2020) [b](Jaderberg et al., 2015) [c](Wang et al., 2020) [d](Lippe et al., 2020) [f](Jaderberg et al., 2014) [g](Wang et al., 2018) [h](Xie et al., 2020) [i](Mahajan et al., 2018)

H.2. Optical Character Recognition

Although visualizations have shown that ImageNet models contain features that respond to the presence of text in an image (Zeiler & Fergus, 2014), these representations are not sufficiently fine-grained to use for the task of optical character recognition (OCR). To compensate, models are augmented with the outputs of custom OCR engines and features to boost performance on tasks where this capability is required (Singh et al., 2019; Yang et al., 2020). Early during the development of CLIP, we noticed that CLIP began to learn primitive OCR capabilities which appeared to steadily improve over the course of the project. To evaluate this qualitatively noticed behavior, we measured performance on 5 datasets requiring the direct and indirect use of OCR. Three of these datasets MNIST (LeCun), SVHN (Netzer et al., 2011), and IIIT5K (Mishra et al., 2012) directly check the ability of a model to perform low-level character and word recognition, while Hateful Memes (Kiela et al., 2020) and SST-2 (Socher et al., 2013b) check the ability of a model to use OCR to perform a semantic task. Results are reported in Table 14.

CLIP's performance is still highly variable and appears to be sensitive to some combination of the domain (rendered or natural images) and the type of text to be recognized (numbers or words). CLIP's OCR performance is strongest Hateful Memes and SST-2 - datasets where the text is digitally rendered and consists mostly of words. On IIIT5K, which is natural images of individually cropped words, zero-shot CLIP performs a bit more respectively and its performance is similar to Jaderberg et al. (2014) early work combining deep learning and structured prediction to perform open-vocabulary OCR. However, performance is noticeably lower on two datasets involving recognition of hand written and street view numbers. CLIP's 51% accuracy on full number SVHN is well below any published results. Inspection suggests CLIP struggles with repeated characters as well as the low resolution and blurry images of SVHN. CLIP's zero-shot MNIST performance is also poor and is outperformed by supervised logistic regression on raw pixels, one of the simplest possible machine learning baselines.

SST-2 is a sentence level NLP dataset which we render into images. We include SST-2 in order to check whether CLIP is able to convert low level OCR capability into a higher level representation. Fitting a linear classifier on CLIP's representation of rendered sentences achives 80.5% accuracy. This is on par with the 80% accuracy of a continuous bag of words baseline using GloVe word vectors pre-trained on 840 billion tokens (Pennington et al., 2014). While this is a

simple NLP baseline by today's standard, and well below the 97.5% of the current SOTA, it is encouraging to see that CLIP is able to turn an image of rendered text into a non-trivial sentence level representation. Fully supervised CLIP is also surprisingly strong on Hateful Meme detection, where CLIP is only 0.7 points behind the current single model SOTA and several points above the best baseline from the original paper. Similar to SST-2, these other results on Hateful Memes use the ground truth text which CLIP does not have access to. Finally, we note that zero-shot CLIP outperforms the best results using fully supervised linear probes across all other 56 models included in our evaluation suite. This suggests CLIP's OCR capability is at least somewhat unique compared to existing work on self-supervised and supervised representation learning.

		UCF101	K700	RareAct	
		Top-1	AVG	mWAP	mWSAP
Finetune	R(2+1)D-BERT[a]	**98.7**	-	-	-
	NS ENet-L2[b]	-	**84.8**	-	-
	HT100M S3D[d]	91.3	-	-	-
	Baseline I3D[e]	-	70.2	-	-
Linear	MMV FAC[f]	91.8	-	-	-
	NS ENet-L2[c]	89.4[c]	68.2[c]	-	-
	CLIP	92.0	73.0	-	-
ZS	HT100M S3D[d]	-	-	30.5	34.8
	CLIP	80.3	69.6	**40.7**	**44.8**

Table 15. Action recognition performance on 3 video datasets. Single model SOTA reported to best of knowledge. Note that *linear CLIP* and *linear NS ENet-L2* are trained and evaluated on a single frame subsampled version of each dataset and not directly comparable to prior work. On Kinetics-700, we report the ActivityNet competition metric which is the average of top-1 and top-5 performance. [a](Kalfaoglu et al., 2020) [b](Lu et al., 2020) [c](Xie et al., 2020) [d](Miech et al., 2020b) [e](Carreira et al., 2019) [f](Alayrac et al., 2020)

H.3. Action Recognition in Videos

For the purpose of learning, a potentially important aspect of natural language is its ability to express, and therefore supervise, an extremely wide set of concepts. A CLIP model, since it is trained to pair semi-arbitrary text with images, is likely to receive supervision for a wide range of visual concepts involving both common and proper nouns, verbs, and adjectives. ImageNet-1K, by contrast, only labels common nouns. Does the lack of broader supervision in ImageNet result in weaker transfer of ImageNet models to tasks involving the recognition of visual concepts that are not nouns?

To investigate this, we measure and compare the performance of CLIP and ImageNet models on several video action classification datasets which measure the ability of a model to recognize verbs. In Table 15 we report results on UCF-101 (Soomro et al., 2012) and Kinetics-700 (Carreira et al., 2019), two common datasets for the task. Unfortunately, our CPU based linear classifier takes a prohibitively long time to evaluate on a video dataset due to the very large number of training frames. To deal with this, we aggressively sub-sample each video to only a single center frame, effectively turning it into an image classification dataset. As a result, our reported performance in a linear evaluation setting likely under estimates performance by a moderate amount.

Despite this handicap, CLIP features transfer surprisingly well to this task. CLIP matches the best prior result on UCF-101 in a linear probe evaluation setting and also outperforms all other models in our evaluation suite. On Kinetics-700, CLIP also outperforms the fine-tuned I3D baseline from the original paper. Since it does not require a training stage, we report CLIP's zero-shot performance when averaging predictions across all frames. CLIP also performs well in this setting and on Kinetics-700 its performance is within 1% of the fully supervised I3D baseline which is trained on 545000 labeled videos. Encouraged by these results, we also measure CLIP's performance on the recently introduced RareAct dataset (Miech et al., 2020a) which was designed to measure zero-shot recognition of unusual actions like "hammering a phone" and "drilling an egg". CLIP improves over the prior state of the art, a S3D model trained on automatically extracted captions from 100 million instructional videos, by 10 points.

While CLIP has encouragingly strong performance on the task of action recognition, we note that there are many differences between the models being compared beyond just their form of supervision such as model architecture, training data distribution, dataset size, and compute used. Further work is needed to more precisely determine what specific design decisions contribute to achieving high performance on this task.

H.4. Geolocalization

Another behavior we noticed during the development of CLIP was its ability to recognize many places and locations. To quantify this we created the Country211 dataset as described in Appendix A and report results on it throughout the paper. However it is a new benchmark so to compare with prior work on geolocalization we also report results on the IM2GPS test set from Hays & Efros (2008) in Table 17. Since IM2GPS is a regression benchmark, we guess the GPS coordinates of the nearest image in a set of reference images using CLIP's embedding space. This is not a zero-shot result since it uses nearest-neighbor regression. Despite querying only 1 million images, which is much less than prior work, CLIP performs similarly to several task specific models. It is not, however, competitive with the current state

	IN Top-1	IN-V2 Top-1	IN-A Top-1	IN-R Top-1	ObjectNet Top-1	IN-Sketch Top-1	IN-Vid PM0	IN-Vid PM10	YTBB PM0	YTBB PM10
NS EfficientNet-L2[a]	**88.3**	**80.2**	**84.9**	74.7	68.5	47.6	88.0	82.1	67.7	63.5
FixResNeXt101-32x48d V2[b]	86.4	78.0	68.4	80.0	57.8	59.1	85.8	72.2	68.9	57.7
Linear Probe CLIP	85.4	75.9	75.3	84.2	66.2	57.4	89.1	77.2	68.7	63.1
Zero-Shot CLIP	76.2	70.1	77.2	**88.9**	**72.3**	**60.2**	**95.3**	**89.2**	**95.2**	**88.5**

Table 16. Detailed ImageNet robustness performance. IN is used to abbreviate for ImageNet. [a](Xie et al., 2020) [b](Touvron et al., 2019)

	1km	25km	200km	750km	2500km
ISNs[a]	**16.9**	**43.0**	**51.9**	**66.7**	**80.2**
CPlaNet[b]	16.5	37.1	46.4	62.0	78.5
CLIP	13.9	32.9	43.0	62.0	79.3
Deep-Ret+[c]	14.4	33.3	47.7	61.6	73.4
PlaNet[d]	8.4	24.5	37.6	53.6	71.3

Table 17. Geolocalization performance on the IM2GPS test set. Metric is percent of images localized within a given radius. Models are ordered by average performance. [a](Muller-Budack et al., 2018) [b](Hongsuck Seo et al., 2018) [c](Vo et al., 2017) [c](Weyand et al., 2016)

of the art.

H.5. Robustness to Distribution Shift

Section 3.4 provides a high level summary and analysis of ImageNet-related robustness results. We briefly provide some additional numerical details in this appendix. Performance results per dataset are provided in Table 16 and compared with the current state of the art results reported in Taori et al. (2020)'s evaluation suite. Zero-shot CLIP improves the state of the art on 5 of the 7 datasets, ImageNet-R, ObjectNet, ImageNet-Sketch, ImageNet-Vid, and Youtube-BB. CLIP's improvements are largest on ImageNet-Vid and Youtube-BB due to its flexible zero-shot capability and on ImageNet-R, which likely reflects CLIP's pre-training distribution including significant amounts of creative content. A similar behavior has been documented for the Instagram pre-trained ResNeXt models as discussed in Taori et al. (2020).

I. Model Hyperparameters

Hyperparameter	Value
Batch size	32768
Vocabulary size	49408
Training epochs	32
Maximum temperature	100.0
Weight decay	0.2
Warm-up iterations	2000
Adam β_1	0.9
Adam β_2	0.999 (ResNet), 0.98 (ViT)
Adam ϵ	10^{-8} (ResNet), 10^{-6} (ViT)

Table 18. Common CLIP hyperparameters

Model	Learning rate	Embedding dimension	Input resolution	ResNet blocks	ResNet width	Text Transformer layers	Text Transformer width	Text Transformer heads
RN50	5×10^{-4}	1024	224	(3, 4, 6, 3)	2048	12	512	8
RN101	5×10^{-4}	512	224	(3, 4, 23, 3)	2048	12	512	8
RN50x4	5×10^{-4}	640	288	(4, 6, 10, 6)	2560	12	640	10
RN50x16	4×10^{-4}	768	384	(6, 8, 18, 8)	3072	12	768	12
RN50x64	3.6×10^{-4}	1024	448	(3, 15, 36, 10)	4096	12	1024	16

Table 19. CLIP-ResNet hyperparameters

Model	Learning rate	Embedding dimension	Input resolution	Vision Transformer layers	Vision Transformer width	Vision Transformer heads	Text Transformer layers	Text Transformer width	Text Transformer heads
ViT-B/32	5×10^{-4}	512	224	12	768	12	12	512	8
ViT-B/16	5×10^{-4}	512	224	12	768	12	12	512	8
ViT-L/14	4×10^{-4}	768	224	24	1024	16	12	768	12
ViT-L/14-336px	2×10^{-5}	768	336	24	1024	16	12	768	12

Table 20. CLIP-ViT hyperparameters