This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Teaching CLIP to Count to Ten

Roni Paiss[1,2] Ariel Ephrat[1] Omer Tov[1] Shiran Zada[1]
Inbar Mosseri[1] Michal Irani[1,3] Tali Dekel[1,3]
[1]Google Research [2]Tel Aviv University [3]Weizmann Institute of Science

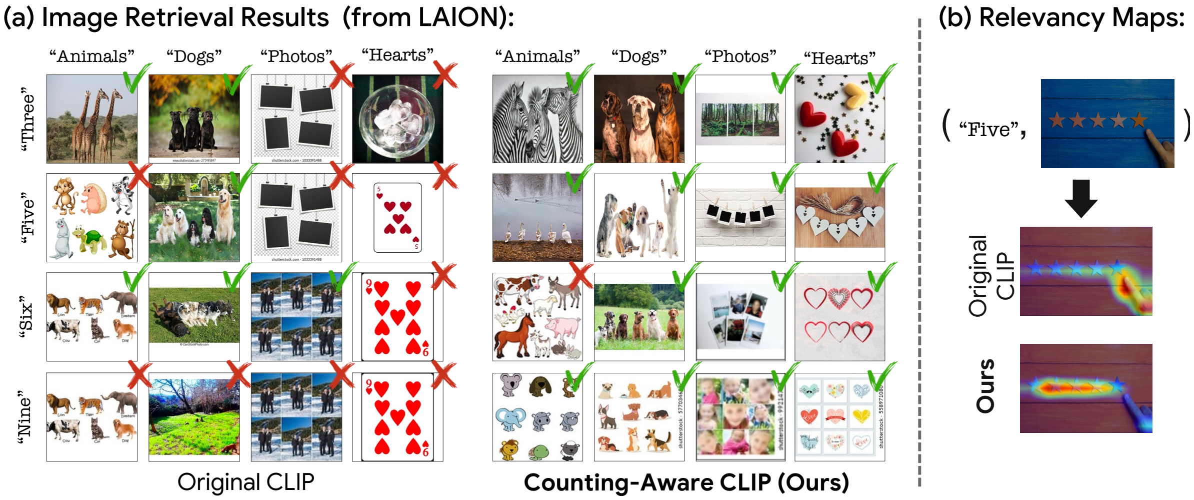


Figure 1: **Inability of CLIP to count.** *We show that CLIP is insensitive to the number of objects in an image, and demonstrate the effectiveness of our Counting-Aware CLIP. (a) Image retrieval results using text captions of the form "a photo of <number> <objects>", with different numbers and types of objects. Images that match the caption are marked with ✓; images that do not are marked with ✗. Our Counting-Aware model retrieves images that depict the requested number of objects, while the original CLIP often retrieves images that contain the wrong number of objects, or images where the number is explicitly written in the image (e.g. "nine hearts" - the image contains the number "9", but has 11 hearts). (b) Attention maps demonstrating that our model attends to all matching object instances in the image, unlike the original CLIP.*

Abstract

Large vision-language models (VLMs), such as CLIP, learn rich joint image-text representations, facilitating advances in numerous downstream tasks, including zero-shot classification and text-to-image generation. Nevertheless, existing VLMs exhibit a prominent well-documented limitation – they fail to encapsulate compositional concepts such as counting. We introduce a simple yet effective method to improve the quantitative understanding of VLMs, while maintaining their overall performance on common benchmarks. Specifically, we propose a new counting-contrastive loss used to finetune a pre-trained VLM in tandem with its original objective. Our counting loss is deployed over automatically-created counterfactual examples, each consisting of an image and a caption containing an incorrect object count. For example, an image depicting three dogs is paired with the caption "Six dogs playing in the yard" as a negative example. Our loss encourages discrimination between the correct caption and its counterfactual variant which serves as a hard negative example. To the best of our knowledge, this work is the first to extend CLIP's capabilities to object counting. Furthermore, we introduce "CountBench" – a new image-text counting benchmark for evaluating object counting capabilities. We demonstrate a significant improvement over state-of-the-art baseline models on this task. Finally, we leverage our counting-aware CLIP model for image retrieval and text-conditioned image generation, demonstrating that our model can produce specific counts of objects more reliably than existing ones.

1. Introduction

Since the advent of CLIP [38], training large vision-language models (VLMs) has become a prominent paradigm for representation learning in computer vision. By observing huge corpora of paired images and captions crawled from the Web, these models learn powerful and rich joint image-text embedding spaces, which have been employed in numerous visual tasks, including classification [58, 59], segmentation [27, 53], motion generation [47], image captioning [30, 48], text-to-image generation [12, 29, 32, 41, 44] and image or video editing [4, 6, 9, 17, 25, 35, 49, 18, 7]. Recently, VLMs have also been a key component in text-to-image generative models [5, 39, 41, 43], which rely on their textual representations to encapsulate the semantic meaning of the input text.

Despite their power, prominent VLMs, such as CLIP [38] and BASIC [36], are known to possess a weak understanding of the number of objects present in an image [3, 36, 38]. This is demonstrated in Fig. 1, where, when given a caption of the template "a photo of $<number>$ $<objects>$", CLIP often fails to retrieve images that correctly match the described number. Downstream applications that rely on VLM-based representations inherit these limitations, e.g., image generation models struggle to reliably produce specific counts of objects [43, 51, 52].

In this work, we introduce a novel method that enhances the quantitative understanding of large-scale VLMs by encouraging them to produce representations that are sensitive to the number of objects in the image and text.

We hypothesize that the reason existing VLMs fail to learn the concept of counting is severalfold: (i) Captions that accurately specify the number of objects become extremely rare in the data as the number of objects increases. For example, we found that for more than five objects, captions typically contain a general form of quantity, e.g., "a group of..." or "many...", rather than an accurate count. This can be attributed to the fact that people cannot instantly identify numerical quantities larger than four without explicitly counting the objects [24]. (ii) Numbers in the caption often refer to attributes that are NOT related to counting – e.g. age, time, address, temperature, model number ("This is an iPhone 5"), etc. (iii) The task of counting (associating the visible number of objects in an image with the number in the caption), is not explicitly enforced in current VLM training objectives. Therefore, current VLMs are able to count reasonably well only up to two or three (for which there are sufficient image-caption examples).

We thus suggest to mitigate each of these problems by: (i) Creating suitable training data in which the captions contain accurate numbers of objects. (ii) Designing a training objective whereby understanding object counts is critical for discriminating between the correctly associated caption and incorrect ones.

More specifically, as illustrated in Fig. 2, we automatically create a clean and diverse *counting training set* by curating image-text examples where the image depicts multiple objects and its caption is verified to express their count. We then finetune a pretrained VLM by formulating counting as a discriminative task – for each example, we create a counterfactual caption by swapping the spelled number associated with the object count with a different randomly selected number. The model's objective is then to associate the image correctly with its true count caption, discriminating it from the counterfactual one.

To evaluate counting capabilities, we introduce *"CountBench"* – a carefully curated object counting benchmark, consisting of 540 diverse, high quality image-text examples. We evaluate our method on two prominent contrastive VLMs: CLIP [38] and BASIC [36], and demonstrate a significant improvement in accuracy in the task of zero-shot count classification over baseline models. Importantly, we achieve this while maintaining the original knowledge learned by the VLM, as demonstrated by an extensive evaluation of our model on standard zero-shot downstream tasks. The quantitative understanding of our model is further evident by our text-to-image retrieval results (e.g., Fig. 1(a)), as well as by the relevancy maps of our model, which demonstrate that the model correctly attends to all visible objects whose count is specified in the text (e.g., Fig. 1(b)). Finally, we train a large-scale text-to-image generative model [43] which incorporates our counting training set and counting-aware CLIP text encoder. The generated images from this model exhibit higher fidelity to the number of objects specified in the input prompts (Fig. 8).
To summarize, our main contributions are:

1. A novel training framework for tackling the task of vision-language counting – an important limitation of current VLMs.
2. A new benchmark, "*CountBench*", carefully filtered and validated for evaluating VLMs on the counting task.
3. We apply our method to the widely-adopted VLMs CLIP and BASIC, demonstrating significant improvement on the counting task, while maintaining general (non-counting) performance on common benchmarks.
4. We utilize our counting-aware VLMs for downstream tasks including image retrieval and text-to-image generation, demonstrating more reliable results when the text prompt contains a specific number of objects.

2. Related work

Contrastive vision-language models: Vision-language models have demonstrated impressive success in vision and multimodal tasks [2, 11, 36, 38, 46, 54, 37]. In this work, we focus on contrastive VLMs, such as CLIP [38] and BASIC [36], as they are widely used both for downstream tasks and as backbones for generative models [40, 43]. Both are trained with a contrastive objective, where matching text-image pairs should have a low cosine distance, and non-matching texts and images should be far apart. The representations computed by CLIP have proven to be very effective in vision and multimodal tasks, due to their zero-shot

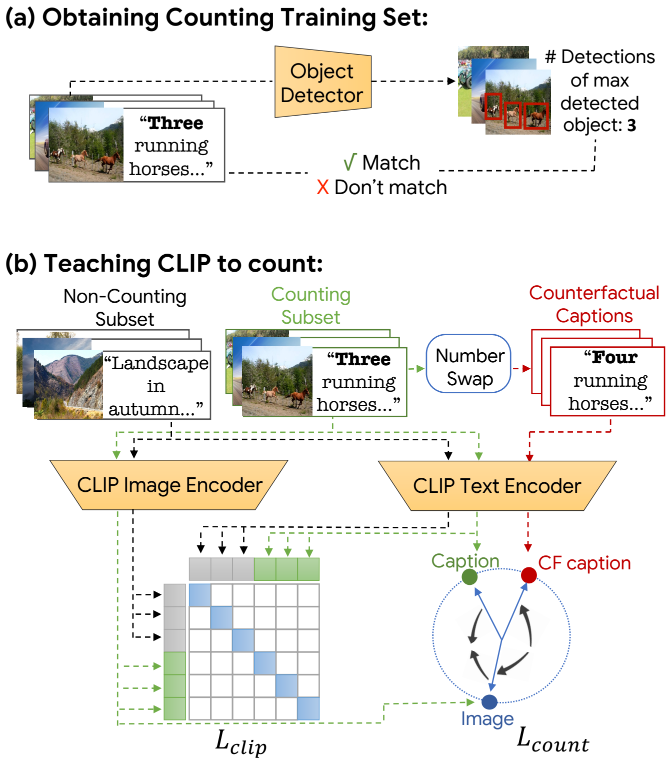


Figure 2: **Method overview** *(a) We create a text-image counting training set in which each caption expresses the number of objects depicted in the corresponding image. This is done by using an off-the-shelf object detector to automatically identify text-image examples in which the text count matches the number of visible objects in the image (see Sec. 3.1). (b) We finetune a pre-trained CLIP model using our counting subset (a), through a dedicated contrastive objective L_{count}, used in addition to the original (general) contrastive objective (L_{clip}). Specifically, given a text-image example from our counting subset, we automatically create a counterfactual prompt by replacing the true object count in the original caption with an incorrect count; L_{count} encourages the model to embed the image close to its original caption embedding (expressing the true object count) and far from its counterfactual count (see Sec. 3.2).*

capabilities and semantic nature, and have been used as a prominent component in numerous tasks and methods. BASIC [36] scaled up the size of the model, batch size and dataset, improving zero-shot accuracy on common benchmarks.

Subitizing: "Subitizing" [24] is the ability of people to instantly recognize the number of objects at a glance – without actually counting them. It has been shown [23] that most humans can subitize only a small number of objects (up to ∼4). This cognitive phenomenon is reflected in the numbers people tend to specify in captions (⩽4), and subsequently in the counting abilities of VLMs, which struggle to count larger numbers. Subitizing was also referred to in computer vision [50, 56]. These papers aim to predict a very small number (⩽4) of salient objects in an image, and are pure image-based (do not involve any text or VLMs).

Counting in vision-language models: While demonstrating impressive recognition capabilities, large VLMs such as CLIP[38] and BASIC [36] are known to only partially capture the meaning of the text. Thus, they fail to understand the number of objects in an image [3, 36, 38]. Paiss et al. [34] demonstrated that CLIP attends to only a small subset of its input, mainly the nouns, and often ignores adjectives, numbers and prepositions.

Counting has remained a stand-alone task under the domain of visual question answering (VQA), tackled with specifically designed architectures and techniques [1, 31, 57]. Our work defers from these prior efforts in several key aspects: (i) While previous efforts are restricted to VQA architectures and problem formulation, our goal is to improve the quantitative understanding of general-purpose contrastive VLMs (e.g., CLIP and BASIC), used in various vision and multimodal tasks where counting-aware solutions are not currently available. (ii) Our work can enhance the zero-shot counting capabilities of VLMs to unrestricted objects, unlike prior methods that are trained on specific domains, which can be problematic for new domains where no counting labels are available.

3. Method

Our goal is to teach a pre-trained VLM (e.g., CLIP) to count, i.e., to improve its quantitative textual and visual understanding. Our framework, illustrated in Fig. 2, consists of two main stages. We first automatically create a *counting training set*, comprising clean and diverse images along with corresponding captions that describe the number of visible objects in the scene. We then leverage this dataset to finetune the VLM through a designated count-based contrastive loss that is used in tandem with the original generic image-text objective.

More specifically, our key idea is to automatically generate counterfactual examples by swapping the true object count in the caption with a different random number. Our new counting loss encourages the model to embed an image close to its true count, as expressed by the original caption, while pushing it away from the embedding of the counterfactual count prompt. As the only difference between the correct caption and its counterfactual counterpart is a single word—the spelled number of objects—the model has to distinguish between the correct and incorrect count in order to succeed in its training task. Next, we describe our dataset creation and finetuning paradigm in detail.

3.1. Creating an image-text counting train set

A naïve approach for obtaining an image-text counting dataset is to filter a large-scale dataset for examples in which the caption contains a number. However, this approach results in a highly noisy dataset, since the numbers in the captions often refer to other numerical attributes that are unrelated to object counts. Such numerical attributes are age ("A 7 year old girl"), time ("The time is 9 o'clock"), addresses, model numbers, etc (see examples in supplementary). To ensure that the numbers in the captions indeed specify object counts, we employ several stages of automatic filtering in our data pipeline (Fig. 2 (a)):

First, we filter out all examples whose caption does not contain a spelled number $\in \{\text{“}two\text{”}, \ldots, \text{“}ten\text{”}\}$. We do so, as we observed that non-spelled numbers, or numbers higher than ten, mostly appear in conjunction with a measure of time, (e.g. dates) or addresses, rather than numbers of objects present in the image.

In the second stage, we verify that the spelled numbers serve as object counters, and that the counted objects are visible and detectable in the images. For example, for the caption "A photo of *three* dogs", we verify that the image indeed depicts three visible dogs, no more, and no less. This count verification is achieved automatically by first applying an off-the-shelf object detector [22], and counting the number of detections per object. We assume that the caption refers to the most prevalent object in the image. Thus, we retain only examples for which the number specified in the caption aligns with the number of instances of the maximally-detected object. We denote by C our automatically filtered train set.

Naturally, the filtered data C is imbalanced. The number of examples that pass our filtering drops significantly as the count increases, e.g., the number of "ten" image-text pairs is around $1000\times$ smaller than "two". Training with such imbalanced data creates a bias—the loss can be reduced by classifying frequent numbers as the correct caption and rare numbers as counterfactual, regardless of the image content. Therefore, balancing the data is of essence. Due to scarcity of examples depicting more than six objects, we choose to balance the numbers "two" – "six" separately from the higher numbers "$seven$" – "ten". For each of the numbers "two" – "six", we sample around $37K$ samples, while for "$seven$" – "ten", we use all the samples passed by our filter. There are approximately $7K$ samples for "$seven$" down to around $1.5K$ samples for "ten". We found this approach to provide us with a diverse and relatively balanced training dataset, yet more sophisticated methods could be considered in the future. From this point on, C will denote our filtered and balanced numbered training set.

3.2. Teaching CLIP to count

Our goal is to improve the quantitative understanding of a pre-trained VLM (e.g., CLIP), while preserving its real-world knowledge, as reflected by its zero-shot capabilities on commonly-evaluated benchmarks. Therefore, we use a combination of two loss functions:

$$L = L_{CLIP} + \lambda L_{count} \tag{1}$$

where L_{CLIP} is the regular contrastive loss of CLIP, L_{count} is our counting-designated loss (described below), and λ is a hyperparameter used to weight the two losses.

We finetune the model on two training sets: (i) A very large dataset collected from the Web that contains general in-the-wild images and captions. (ii) Our filtered numbered training set C, described in Sec. 3.1, which contains samples where object counts are spelled out in the captions. While L_{CLIP} is calculated on all samples, the counting loss L_{count} is calculated only on samples from C. For each image-text pair $(i_k, t_k) \in C$, a counterfactual caption t_k^{CF} is automatically created by swapping the number in the caption t_k with a different random number (e.g., the caption "five dogs" can be counterfactualized with "eight dogs"). At each training step, the triplets $(i_k, t_k, t_k^{CF})_{k=1}^N$ are then fed to CLIP's text and image encoders to obtain their embeddings $(ei_k, et_k, et_k^{CF})_{k=1}^N$.

Then, a contrastive loss L_{count} is computed to enforce that the similarity score of the image is high with the original caption and low with the counterfactual caption:

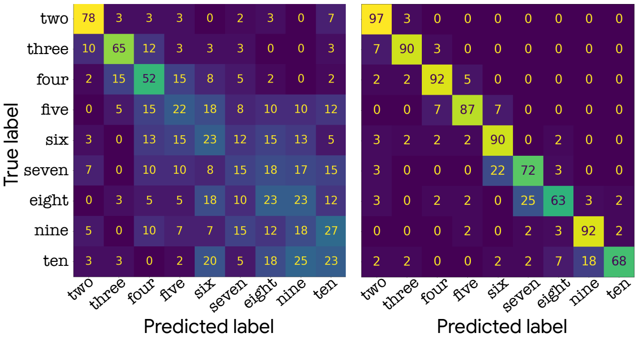
$$L_{count} = -\frac{1}{N}\sum_{k=1}^{N} \log \frac{\exp(ei_k \cdot et_k)}{\exp(ei_k \cdot et_k) + \exp(ei_k \cdot et_k^{CF})} \tag{2}$$

Since the original ground truth caption and counterfactual caption differ only by the number of objects specified in them, this loss encourages the model to learn the relationship between the specified spelled number and the number of the objects it refers to.

We use the negative samples only in the counting objective L_{count}, instead of adding them to the batch for the contrastive loss L_{CLIP} in order to better weight their effect.

4. "CountBench" – an evaluation benchmark

We introduce a new object-counting evaluation benchmark called *CountBench*. This benchmark was automatically curated (and manually verified) from the publicly available LAION-400M image-text dataset [45]. CountBench contains a total of 540 images containing between 2 and 10 instances of a particular object, where their corresponding captions reflect this number. This benchmark is used only for testing and evaluation. ***CountBench has no overlap with our training set C.***



(a) Baseline CLIP (L/14)

(b) Ours (L/14)

Figure 3: **Confusion matrices on CountBench.** *Classification accuracy on CountBench, broken down into confusion matrices for the public CLIP ViT-L/14 (a), and our counting-aware CLIP ViT-L/14 model (b), demonstrating clear quantitative superiority of our model.*

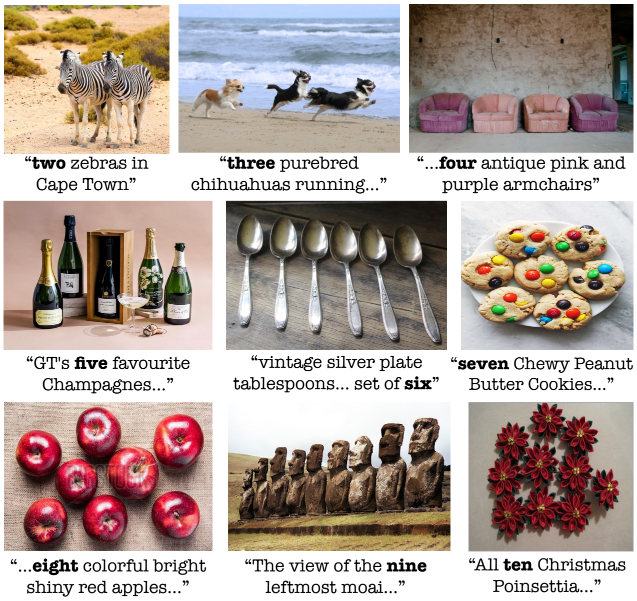
The images in *CountBench* were obtained by running our automatic filtering method described in Sec. 3.1 on the entire LAION-400M dataset. The filtering pipeline includes applying an object detector, whose benefits are twofold: (i) it facilitates verification of the object count specified in the caption, and (ii) it ensures that the counted objects are visible and detectable. The latter is important since benchmarks that do not disentangle the problem at hand from other challenges (e.g. occlusions and low resolution) may not reflect the performance on the actual task, since the difficulty can often stem from unrelated factors [14].

This filtering produced over 158K images for the number "*two*", but only around 100 images for "*ten*", demonstrating again the severe number imbalance we encountered with our training set. After automatically balancing each number to 100-200 samples each, the entire dataset was manually verified to contain only pairs in which the spelled number in the caption matches the number of clearly visible objects in the image. The dataset was rebalanced after this stage, ending up with 60 image-text pairs per number $\in \{$"*two*", .., "*ten*"$\}$, 540 in total. Samples from the dataset can be seen in Fig. 4.

We use the *CountBench* benchmark to evaluate the counting abilities of the models trained with our method in Sec. 5.

5. Experiments

We thoroughly evaluate our method, both quantitatively and qualitatively, on object counting-related tasks using our *CountBench* benchmark. We further validate that the performance of our counting-aware models on a variety of *general* zero-shot classification benchmarks is retained [8, 13, 15, 16, 19, 20, 21, 26, 33, 42, 55]. To gain a better understanding of our models, we show visualizations of text-image relevancy maps, along with per-word relevancy scores, demonstrating that our model indeed attends to the number of objects in the image and text. Finally, we apply our model to text-to-image retrieval and generation, producing specific numbers of objects more reliably than baseline models.



"**two** zebras in Cape Town"

"**three** purebred chihuahuas running..."

"...**four** antique pink and purple armchairs"

"GT's **five** favourite Champagnes..."

"vintage silver plate tablespoons... set of **six**"

"**seven** Chewy Peanut Butter Cookies..."

"...**eight** colorful bright shiny red apples..."

"The view of the **nine** leftmost moai..."

"All **ten** Christmas Poinsettia..."

Figure 4: **CountBench benchmark.** *Sample images and their corresponding captions from our new CountBench object counting benchmark. This benchmark was automatically curated (and manually verified) from the publicly-available LAION-400M dataset.*

5.1. Zero-shot counting accuracy

We evaluate our models and baselines on *CountBench* on the task of classifying the number of objects in an image in a zero-shot manner. For each image in *CountBench* we augment the existing caption with eight other possible captions by replacing the number in its caption with all the numbers $\in \{$"*two*", . . . , "*ten*"$\}$, and calculate the similarity score between the image and each of the nine captions. The number in the caption that obtains highest similarity score with the image is considered the predicted number.

Table 1 reports the results of this evaluation on two prominent contrastive VLMs: CLIP-B/32 and BASIC-S. We report both the counting accuracy (selection of the correct number) and the mean deviation of the models' predictions from the correct numbers. For each of the architectures, we compare our model (configuration E) with two baseline configurations: (A) the official baseline model, and (B) the baseline model finetuned on our general text-image dataset used in our implementation, with the standard contrastive loss. Comparing the performance of these config-

	CLIP-B/32					BASIC-S				
	A Official Baseline	B Internal Baseline	C Ours (w/o L_{count})	D Ours (Naive Filtering)	E Ours	A Official Baseline	B Internal Baseline	C Ours (w/o L_{count})	D Ours (Naive Filtering)	E Ours
Accuracy ↑	31.67	32.94	44.26	49.81	**75.93**	17.97	22.75	30.59	28.68	**69.02**
Mean deviation from the correct number ↓	1.53	1.44	0.97	1.28	**0.49**	2.13	2.02	1.29	1.87	**0.64**

Table 1: **Quantitative counting results.** *Top-1 zero-shot accuracy and the mean absolute distance between the predicted numbers and the true numbers on CountBench. We compare several configurations: (A) The official CLIP [38] and BASIC [36] models. (B) The official baselines finetuned on our internal curated data. (C) Models trained with our filtered counting set, without L_{count} (D) Models finetuned with L_{count} on a naively filtered counting set (E) Our method, which is significantly superior to all other configurations, both in accuracy and deviation from correct number.*

	CLIP-B/32			BASIC-S		
	Official Baseline	**Internal Baseline**	**Ours**	**Official Baseline**	**Internal Baseline**	**Ours**
ImageNet	62.93	64.97	64.06	59.70	61.96	61.18
CIFAR10	63.91	61.00	60.65	76.22	84.69	84.05
CIFAR100	33.10	32.49	33.56	45.35	56.80	55.89
Caltech101	75.99	82.50	82.36	78.16	81.03	81.05
EuroSAT	45.23	41.66	37.69	28.39	45.82	45.97
Food101	83.08	80.72	80.53	77.08	77.80	77.06
ImageNetA	31.85	30.85	29.81	17.65	22.55	21.68
ImageNetR	69.38	70.17	70.30	67.11	67.68	66.95
ImageNetV2	55.65	56.56	56.62	52.22	54.35	53.60
Oxford Pets	87.35	87.74	87.41	80.62	85.15	84.87
Oxford Flowers	66.14	65.73	67.39	64.74	66.40	65.90

Table 2: **Zero-shot accuracy on common benchmarks.** *We compare the zero-shot accuracy of our method and baselines on a variety of popular benchmarks. As seen, our method preserves the performance of the original model.*

urations allows us to quantify the effect of using our own large-scale text-image dataset, which differs from the original unpublished data the models were trained on.

As can be seen, our method (*E*) achieves significantly superior counting accuracy compared to the baselines (*A*, *B*), attaining 2–3× higher counting accuracy and more than 3× lower mean deviation from the correct number. Results for different categories in *CountBench* are reported in App. 1.

Tab. 1 also contains an ablation of the two components of our method: filtering a counting training set and finetuning with an additional loss L_{count}. Models with configuration *C* are finetuned on the filtered subset with no counting loss. The large gap in accuracy on *CountBench* between configurations *C* and *E* shows the importance of our loss for the improvement in counting abilities. Models with configuration *D* are finetuned with the counting loss L_{count} on an alternative counting subset, which consists of all the samples that contain spelled numbers $\in$ {"*two*", .., "*ten*"} without additional filtering. The significant difference in counting accuracy between configurations *D* and *E* demonstrates the importance of our restrictive filtering pipeline, as both configurations are finetuned with L_{count} over the samples from a dedicated counting training set. As can be seen in Tab. 1, while the naively filtered data does improve performance over a baseline trained without a dedicated counting subset, the obtained results are still significantly lower than those produced by our model. We attribute this gap in performance to mislabeled training samples in the naively filtered data, which are absent from our counting training set C due to our filtering pipeline.

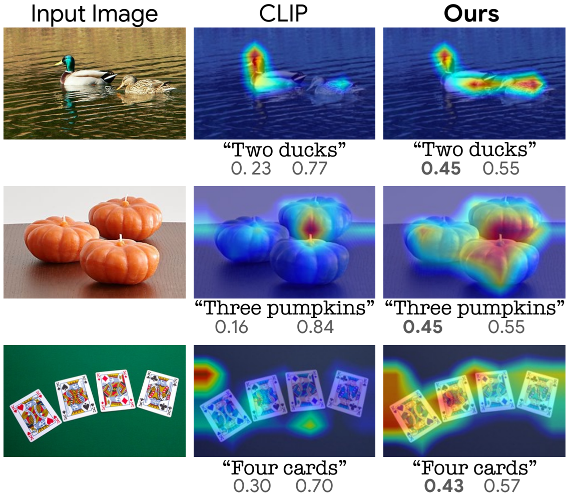


Figure 5: **Relevancy maps of both image and text.** *Visualization of the relevancy maps of both image and text, which represent, for each patch in the image and token in the text, how important it is to the prediction. Using our counting-aware CLIP model, the relevancy of the number (e.g., "four") in the text is increased. In addition, the model focuses on areas in the image that are relevant for counting.*

Confusion matrices for the counting evaluation described above are shown in Fig. 3. For this experiment, we compare a public CLIP-L/14 model against our counting-aware version of it. As can be seen, our counting-aware CLIP model is significantly superior to the baseline across all numbers. It is worth noting that the baseline CLIP performs reasonably well for numbers within the subitizing range ($2 - 4$), but fails on larger numbers. This supports our claim that curated image captions with numbers beyond this range are rare, since specifying them requires manually counting the objects.

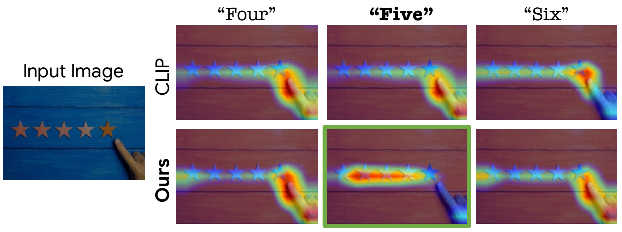


Figure 6: **Relevancy maps for similarity between the image and different numbers.** *We compare the relevancy map of the input image with text prompts of several numbers (i.e. four to six) for both the baseline CLIP and our counting-aware CLIP. Our model focuses on the five stars when calculating the similarity with the prompt "five".*

5.2. Count-based image retrieval

We consider the task of text-to-image retrieval where the text explicitly describes the desired count of objects. To obtain a diverse dataset that consists of varied numbers of objects, yet facilitates retrieval in reasonable time, we split the public LAION-400M dataset [45] into coarse categorical subsets by filtering samples where the caption contains a certain word (e.g., "dogs", "animals", "cars"), and perform retrieval on each of these subsets separately.

For each category, we use the caption "a photo of $<number>$ $<objects>$" where $number \in \{``two", .., ``ten"\}$ (e.g. "a photo of six dogs"). For each caption, we retrieve the three images in the dataset that are predicted by the model to be most similar to the caption. Note that since there are no ground truth labels for the counts of objects, we present qualitative results. Fig. 7 shows the retrieved images using the original CLIP model and our counting-aware CLIP model. An extended figure can be found in the supplementary. As can be seen, when the requested number is larger than three, the images retrieved by the baseline model often depict arbitrary numbers of objects. Additionally, the baseline often retrieves the same images for several different requested numbers. This further implies that the baseline model mostly focuses on the existence of the described object in the image, and ignores the number in the caption. In contrast, our results depict accurate object counts in most cases.

5.3. Performance on non-counting classification

To verify that our counting-aware models preserve the powerful image-text representation capabilities of the original models, we evaluate the zero-shot performance of our models on a variety of common classification benchmarks. Table 2 reports the zero-shot accuracy of our counting aware models against the baselines (corresponding to configurations *A, B* in Tab. 1). As can be seen, our models maintain similar overall accuracy. Additionally, comparing the official baseline and the internal baseline indicates that finetuning the models on our general text-image datasets leads to only a slight shift in the accuracy of the models on common benchmarks. An additional evaluation on the task of general zero-shot image retrieval is reported in the supplementary.

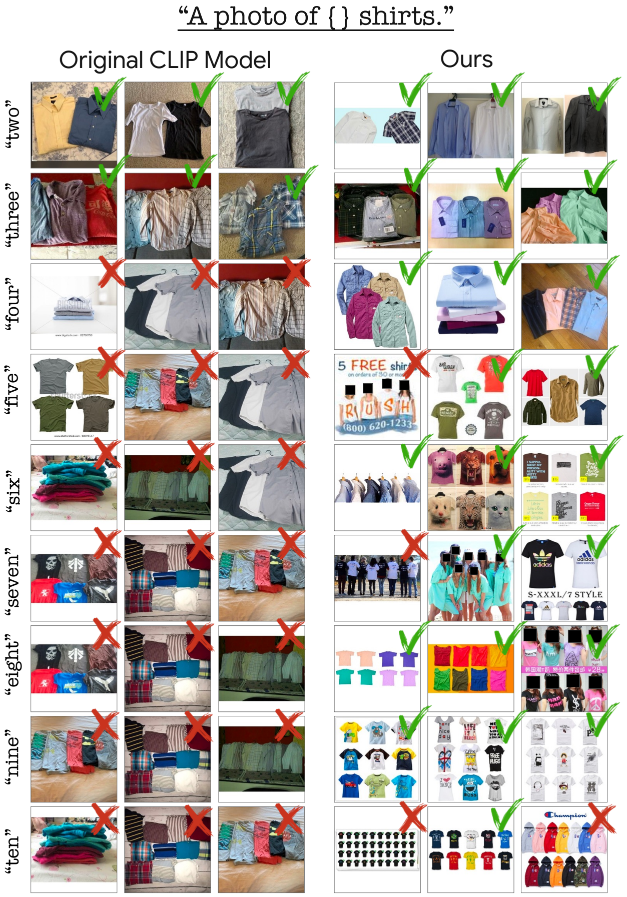


Figure 7: **Top-3 count-based image retrieval** *Text-to-image retrieval results for different counts of shirts, ordered from left to right according to their similarity score (descending). Images that match the caption are marked with ✓ and images that do not match it are marked with ✗. Extended results are in the supplementary.*

5.4. Relevancy map visualization

To gain a better understanding of what our model learns, we use the explainability method of Chefer et al. [10] to visualize relevancy maps, which indicate the importance of different parts of the text and image to the model's similarity score prediction. Fig. 5 displays the normalized relevancy maps of several image-text pairs. Examining the relevancy maps of the text, it is apparent that the relevancy score of the spelled number in the caption is significantly higher for our model than the baseline model, which suggests that our model concentrates more on the mentioned number than the original one. Additionally, examining the relevancy maps of the images, it is evident that our model focuses on all pertinent objects in the image, whereas the original model

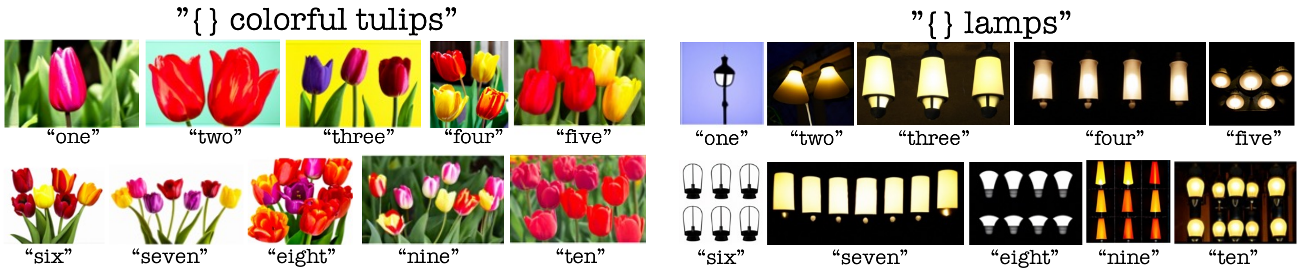


Figure 8: **Generated samples with Imagen using our counting-aware CLIP as backbone**. *The Imagen model benefits from the counting-aware representations produced by the our CLIP model, and is able to generate images that accurately follow the amounts specifies in the captions.*

primarily identifies a single instance of the described object.

To verify that our model does not simply attend to *all* objects that appear in the image, we examined the relevancy maps in Fig. 6 using negative prompts (*i.e*. "four" when there are five elements in the image). Our model focuses only on relevant objects when the correct number is used, unlike the baseline CLIP that highlights all object types in the image. This demonstrates that our model learns to associate the number in the caption with the suitable number of objects, and does not exploit shortcuts or undesired content.

5.5. Count-based text-to-image generation

Text-to-image generative models that rely on CLIP for representations, are known to inherit its limitations [43, 52], thus fail to reliably produce specific counts of objects. In order to demonstrate the effectiveness of our counting-aware model for downstream image generation tasks, we train an Imagen [43] model conditioned on the textual embeddings of a pretrained CLIP-L/14, and another Imagen model conditioned on our counting-aware version of this model. To compare our model and the baseline, we synthesize 12 samples for each prompt in the counting category of the DrawBench benchmark [43]. For each sample, we check whether or not it contains the requested number of objects, as stated in its prompt.

The Imagen model trained with the baseline CLIP achieves a binary accuracy of 24.12%, while the Imagen trained with out counting-aware model reaches 40.35% accuracy. See the supplementary for additional evaluations.

Performance on *non-counting* image generation: To verify that the generation quality is preserved for non-counting captions, we conducted a user study. We presented 52 participants with 20 pairs of images, each containing one image generated with an Imagen model trained with the baseline CLIP, and one with our counting-aware CLIP as backbone. The same random prompt from COCO [28] and seed were used for each pair. Participants were asked which image is better in terms of quality and realism. 54.7% of ratings favored our images, suggesting that our model does not degrade generation quality on general text captions.

5.6. Limitations

Our method is limited by the insufficient existence of training data with images containing multiple instances of an object, along with a corresponding caption that correctly spells out this information. The effect of this data scarcity on our method increases with larger numbers (7, 8, etc.), as people tend to use "a group of" or "many" for large numbers of objects, instead of gruelingly counting them. Furthermore, many of the correct training pairs with higher numbers that do exist, contain relatively simplistic 2D collections of objects, as opposed to objects in a real-world scene. This is also the reason why our current method teaches CLIP to count only up to ten. Mitigating the above-mentioned data limitations, and generalizing to counting beyond 10, can possibly be achieved by concatenating multiple images with smaller numbers of objects. This will result in new training examples with more than ten natural-looking objects, and is part of our future work.

6. Conclusions

We present the first method to enhance VLMs with counting capabilities, while maintaining their overall performance on common benchmarks. This is an essential step towards enabling more accurate text-based image retrieval, classification and generation. A new counting-based contrastive loss is used to finetune a pre-trained VLM in tandem with its original objective. Our counting loss is deployed over automatically-created counterfactual examples, each consisting of an image and a caption containing an incorrect object count. We demonstrate a significant improvement over state-of-the-art baseline models in multiple tasks (classification, retrieval and generation) on several different datasets. Furthermore, we introduce *CountBench* – a new image-text counting benchmark for evaluating VLMs' object counting capabilities. This new benchmark contains in-the-wild images and captions, where the object counts are specified in the captions. Finally, our approach is not specific to counting, and can be generalized to other compositional concepts that VLMs fail to learn.

References

[1] Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *AAAI Conference on Artificial Intelligence*, 2018. 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. 2

[3] Jong Wook Kim Gretchen Krueger Sandhini Agarwal Alec Radford, Ilya Sutskever. Clip: Connecting text and images. *https://openai.com/research/clip*, 2021. 2, 3

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18187–18197, 2022. 2

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[6] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. *ArXiv*, abs/2204.02491, 2022. 2

[7] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. 2

[8] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *ECCV*, 2014. 5

[9] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. Image-based clip-guided essence transfer. In *ECCV*, 2022. 2

[10] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 387–396, 2021. 7

[11] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel M. Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model. *ArXiv*, abs/2209.06794, 2022. 2

[12] Katherine Crowson. Vqgan+clip. *https://colab.research.google.com/drive/1L8oL-vLJXVcRzCFbPwOoMkPKJ8-aYdPN*, 2021. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[14] Anuj Diwan, Layne Berry, Eunsol Choi, David F. Harwath, and Kyle Mahowald. Why is winoground hard? investigating failures in visuolinguistic compositionality. *ArXiv*, abs/2211.00768, 2022. 5

[15] Raveen Doon, Tarun Kumar Rawat, and Shweta Gautam. Cifar-10 classification using deep convolutional neural network. *2018 IEEE Punecon*, pages 1–5, 2018. 5

[16] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. 5

[17] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada. *ACM Transactions on Graphics (TOG)*, 41:1 – 13, 2022. 2

[18] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023. 2

[19] Patrick Helber, Benjamin Bischke, Andreas R. Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12:2217–2226, 2019. 5

[20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8349, October 2021. 5

[21] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 5

[22] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 4

[23] William Stanley Jevons. The power of numerical discrimination. *Nature*, 3:281–282, 1871. 3

[24] E. L. Kaufman and Miles W. Lord. The discrimination of visual number. *The American journal of psychology*, 62 4:498–525, 1949. 2, 3

[25] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022. 2

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 5

[27] Boyi Li, Kilian Q. Weinberger, Serge J. Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *ArXiv*, abs/2201.03546, 2022. 2

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 8

[29] Xingchao Liu, Chengyue Gong, Lemeng Wu, Shujian Zhang, Haoran Su, and Qiang Liu. Fusedream: Training-free text-to-image generation with improved clip+gan space optimization. *ArXiv*, abs/2112.01573, 2021. 2

[30] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2

[31] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. Movie: Revisiting modulated convolutions. 2021. 3

[32] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 2

[33] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 5

[34] Roni Paiss, Hila Chefer, and Lior Wolf. No token left behind: Explainability-aided image classification and generation. In *ECCV*, 2022. 3

[35] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 2

[36] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined Scaling for Open-Vocabulary Image Classification. *arXiv preprint arXiv:2111.10050*, Nov. 2021. 2, 3, 6

[37] Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6967–6977, June 2023. 2

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6

[39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[40] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 2

[41] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021. 2

[42] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019. 5

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. 2, 8

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022. 2

[45] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 4, 7

[46] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15617–15629, 2021. 2

[47] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*, 2022. 2

[48] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zero-shot image-to-text generation for visual-semantic arithmetic. *arXiv preprint arXiv:2111.14447*, 2021. 2

[49] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Bachmann, Amit H. Bermano, Daniel Cohen-Or, Amir Roshan Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Trans. Graph.*, 41:86:1–86:11, 2022. 2

[50] Rijnder Wever and Tom F.H. Runia. Subitizing with variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018. 3

[51] Yonghui Wu and David Fleet. How ai creates photorealistic images from text. *https://blog.google/technology/research/how-ai-creates-photorealistic-images-from-text*, 2022. 2

[52] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yin-

fei Yang, Burcu Karagol Ayan, Benton C. Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *ArXiv*, abs/2206.10789, 2022. 2, 8

[53] Nir Zabari and Yedid Hoshen. Semantic segmentation in-the-wild without seeing any segmentation examples. *ArXiv*, abs/2112.03185, 2021. 2

[54] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18123–18133, June 2022. 2

[55] Hui Zhang, Shenglong Zhou, Geoffrey Y. Li, and Naihua Xiu. 0/1 deep neural networks via block coordinate descent. *ArXiv*, abs/2206.09379, 2022. 5

[56] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe L. Lin, Xiaohui Shen, Brian L. Price, and Radomír Mech. Salient object subitizing. *International Journal of Computer Vision*, 124:169–186, 2015. 3

[57] Y. Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. *ArXiv*, abs/1802.05766, 2018. 3

[58] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[59] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 2