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.

LocTex: Learning Data-Efficient Visual Representations from Localized Textual Supervision

Zhijian Liu
MIT

Simon Stent, Jie Li, John Gideon
Toyota Research Institute

Song Han
MIT

https://loctex.mit.edu/

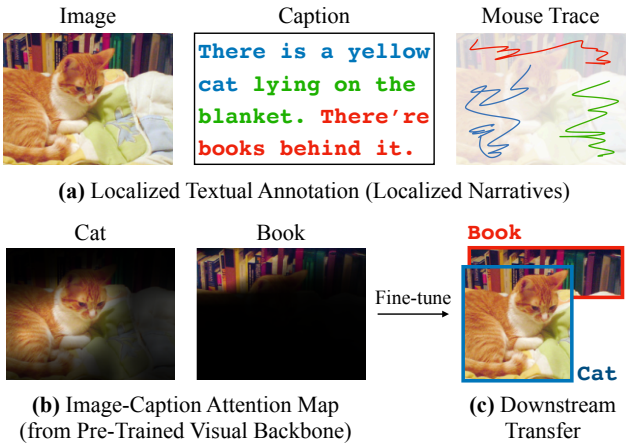
Abstract

Computer vision tasks such as object detection and semantic/instance segmentation rely on the painstaking annotation of large training datasets. In this paper, we propose LocTex that takes advantage of the low-cost ***localized textual annotations*** *(*i.e.*, captions and synchronized mouse-over gestures) to reduce the annotation effort. We introduce a contrastive pre-training framework between images and captions, and propose to supervise the cross-modal attention map with rendered mouse traces to provide coarse localization signals. Our learned visual features capture rich semantics (from free-form captions) and accurate localization (from mouse traces), which are very effective when transferred to various downstream vision tasks. Compared with ImageNet supervised pre-training, LocTex can reduce the size of the pre-training dataset by* ***10****× or the target dataset by* ***2****× while achieving comparable or even improved performance on COCO instance segmentation. When provided with the same amount of annotations, LocTex achieves around* ***4%*** *higher accuracy than the previous state-of-the-art "vision+language" pre-training approach on the task of PASCAL VOC image classification.*

1. Introduction

The tremendous success of deep learning in computer vision can be credited in part to the existence of large annotated datasets, such as ImageNet [7, 47]. However, acquiring high-quality annotations is usually very expensive and time-consuming, especially for dense, pixel-wise labeling tasks. For instance, segmenting instances in a single image from the COCO dataset takes more than 10 minutes on average [29]*.

Pre-training plus fine-tuning is a widely-adopted solution to reduce the need for costly annotations. In the computer vision community, a convolutional neural network (CNN) backbone is first pre-trained to perform image classification on ImageNet. Then, the learned features can be transferred to other downstream tasks by fine-tuning on the target dataset. Over the past few years, this paradigm has enabled state-of-the-art performance on many computer vision tasks, including object detection [46], semantic segmentation [31] and instance segmentation [20].



(a) Localized Textual Annotation (Localized Narratives)

(b) Image-Caption Attention Map (from Pre-Trained Visual Backbone)

(c) Downstream Transfer

Figure 1: LocTex pre-trains the visual CNN backbone with (a) *localized textual annotations*, which consists of free-form captions associated with synchronized mouse traces. With our contrastive and localization loss, the model learns (b) *rich semantics and accurate localization*. This is very useful when transferred to (c) *downstream tasks* that are sensitive to localization (*e.g.*, object detection, instance segmentation).

Though effective, ImageNet pre-training has its caveats. **(i)** Its annotations (*i.e.*, 1000-class labels) are very expensive to acquire. Annotating ImageNet is not as easy as it seems because differentiating among a fine-grained class taxonomy requires expert knowledge, which makes it hard to scale up or repeat. **(ii)** It is not as effective for those tasks that are more sensitive to localization than classification. As ImageNet pre-training only takes the object existence into consideration, its learned visual representations are supposed to be invariant to different object locations. Some recent research [19] has demonstrated competitive performance on object detection and instance segmentation with models trained from scratch.

*70k hours / 320k images = 0.22 hours/image = 13 minutes/image

To solve (i), researchers have explored pre-training backbone networks with coarse, freely available labels, such as metadata and hashtags [23]. There has also been increased attention in self-supervised pre-training that learns visual representations from unlabeled images [18, 5, 16]. Some of them have been successfully scaled up to hundreds of millions or even billions of images [18]. However, (ii) remains unsolved as they usually rely on some low-level visual cues (*e.g.*, color, texture) and lack semantic understanding. In addition to this, **(iii)** self-supervised pre-training methods tend to be trained with prohibitively long schedules to exploit their potential. For instance, the recent approach of BYOL [16] requires 170 TPU days for a single training run.

In this paper, we propose LocTex to learn data-efficient visual representations using *localized textual supervision*, which is composed of free-form captions associated to synchronized mouse traces (see Figure 1a). This form of annotation can be easily acquired from non-expert workers, leading to (i) lower cost and better scalability. Technically, we propose to bridge the vision and language modalities with contrastive learning and supervise the cross-modal attention map with rendered mouse traces, providing (ii) coarse localization information that improves the performance of localization-sensitive downstream tasks. Finally, our method requires (iii) a similar amount of training time as ImageNet pre-training: it can be trained under a day with 8 GPUs.

After the pre-training, we transfer our learned feature representations to various downstream vision tasks, including image classification, object detection and instance segmentation. Compared with the ImageNet supervised pre-training, our proposed LocTex can reduce the size of the pre-training dataset by **10**× or the target dataset by **2**× while achieving comparable or better performance on the COCO instance segmentation. With the same amount of annotations, our LocTex achieves around **4%** higher accuracy than the previous state-of-the-art "vision+language" pre-training approach [8] on the PASCAL VOC image classification.

2. Related Work

Supervised Pre-Training. Much of the recent success of computer vision can be attributed to the richness of image features learned via supervised training. ImageNet pre-training, in which image features are first learned through the supervised image classification on ImageNet [7] before being used on downstream tasks, is a highly popular model initialization method [15, 10]. However, this approach has limitations which have become increasingly evident as the variety of downstream tasks and the types of new annotated data has increased dramatically over the years [44, 19, 61, 51].

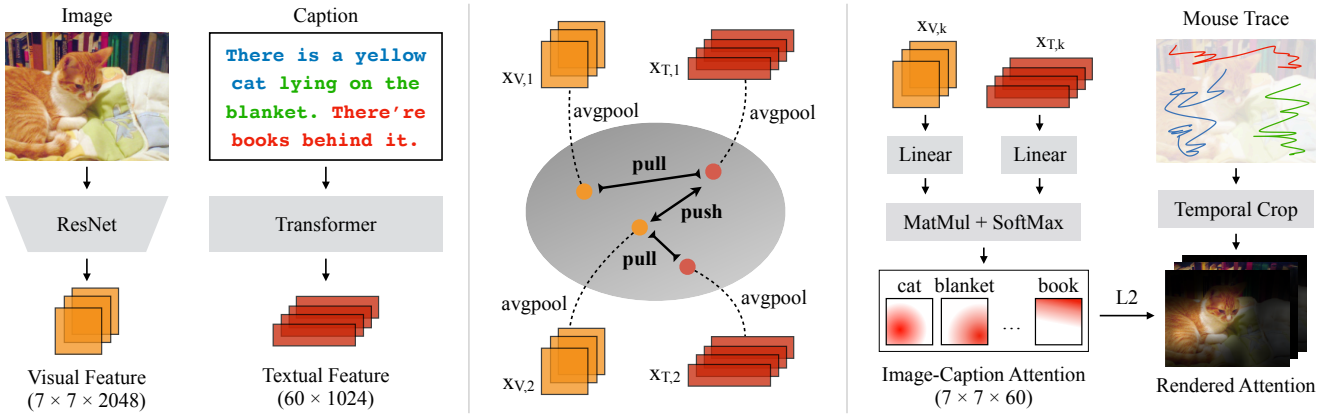
Unsupervised Learning. To go beyond the scale of ImageNet in terms of supervised learning is expensive. Hence, it becomes increasingly popular to seek methods for representation learning that can meet or exceed ImageNet supervised pre-training without the need for labelled data. An important direction in unsupervised learning is "self-supervised" learning, in which models are trained on pretext tasks where training labels can be obtained from the raw or augmented input. Common pretext tasks include predicting context [9], solving jigsaws [34], predicting rotation [14], colorization [58], and inpainting [37]. Generative models have also been widely used in representation learning to reconstruct the distribution of the input data, such as restricted Boltzmann machines (RBMs) [26], autoencoders [25] and generative adversarial networks (GANs) [12, 11]. Recent explorations investigate intra-dataset patterns and feature discrimination, including clustering [3, 4] and contrastive learning [18, 16, 5, 55].

Vision & Language. Pre-training methods in natural language processing have witnessed tremendous improvement over the past few years [6, 39, 43, 2]. Efforts trying to use the text in visual representation learning have never stopped. Early research tried to predict captions or text from associated images [41]. Srivastava *et al*. [52] applied Boltzmann machine to capture multi-modal features. Some works treat text or language as weak supervisory signals for vision and explore the trade-off between label quality and data scale. Li *et al*. [27] train visual models on YFCC-100M [54] using user-provided tags. JFT-300M [53] is also used for visual pre-training with automatic-generated web signals. More recent works like ICMLM [48], VirTex [8] and ConVIRT [59] try to leverage the novel pre-training approaches developed in NLP, such as masked language modeling and transformer-based modeling. A concurrent work of us [42] has explored contrastive learning between image and text at the web scale. In this work, we explore further in this direction with a focus on learning localization-aware features for spatially-sensitive tasks such as object detection and segmentation.

Annotation Efficiency. A key goal of our work is to learn powerful representations on data which can be acquired at a low annotation cost. Recent explorations focused on efficient labeling [1, 30] and active learning [49] approach this problem by placing the model in the loop in order to increase the information gain per unit of annotation effort. We approach from an alternative angle by looking at widely available, natural sources of human supervision, which are already cheap or free to acquire. Our work centers around the Localized Narratives dataset [40], which complements verbal image descriptions with synchronized mouse-over gestures containing noisy spatial cues. We demonstrate that designing a system to leverage such multi-modal cues can provide significant performance benefit on visual representation learning with minimal annotation overhead.

3. Method

In this section, we introduce our approach of visual representation learning from *localized textual supervision*. We



(a) Visual and Textual Backbones **(b)** Contrastive Loss **(c)** Localization Loss

Figure 2: Overview of our data-efficient visual representation learning framework (LocTex). We first use a pair of visual and textual backbones to extract the features from the image and caption. We then apply the contrastive loss to pull the features from positive pairs together and push those from negative pairs apart. Finally, we compute the cross-modal attention map between visual and textual features and provide supervision using the rendered attention from the associated mouse trace.

present an overview of our LocTex framework in Figure 2. We pre-train the visual backbone (as well as the textual backbone) using contrastive learning over positive and negative image-caption pairs. We propose to make use of the accompanying mouse trace annotations to provide coarse learning signals for localization. After pre-training, we transfer the learned visual backbone to other downstream vision tasks (*e.g.*, classification, detection and segmentation).

3.1. Annotations

In the computer vision community, ImageNet [7] was commonly used to pre-train visual backbone networks. However, annotating over 1000 fine-grained classes is very costly and cannot be easily scaled up [48]. In this paper, we propose to employ *localized textual annotations* (also known as localized narratives [40]) as it is relatively cheap to acquire and offers semantically dense information. The annotation we use consists of a caption with synchronized mouse trace:

Caption. Caption is a free-form annotation resulting from annotators being asked to describe the content of the image using natural language. As illustrated in Figure 1a, the information captured in the caption is semantically dense: *i.e.*, the objects in the image as well as their attributes and relative spatial relationships. The underlying rich semantic information could potentially benefit a variety of downstream vision tasks. On the other hand, the cost of this form of annotation is much lower compared with other dense labeling [29] since it is a very natural task for humans to do and does not require the annotator to have extensive training or domain knowledge. Some recent datasets [40] adopt a two-stage data collection pipeline: they first ask the annotators to describe the image verbally and then apply either speech recognition or manual transcription to generate the final caption. From this collection protocol, the starting and ending timestamp of each token can also be obtained (which will be used to synchronize the mouse trace with the caption).

Synchronized Mouse Trace. Compared with drawing a sequence of bounding boxes or instance masks, logging the mouse gestures of the subject while they describe the image is an easier and more natural way for human annotators to specify the object locations. It can be acquired almost freely in the caption annotation pipeline since the annotators only need to additionally hover their mouse over the region being described. Though the localization and semantic correspondence is too coarse for these annotations to be directly used for tasks like object detection, it does capture rich information about "what is where" at a high level.

3.2. Backbones

Given an image and its corresponding caption, we first apply two separate neural networks to extract their features.

Visual Backbone. The visual backbone takes the raw image as input and outputs a feature map that contains the semantic information. This is also the only component which we will transfer to other vision downstream tasks. Theoretically, we can choose any convolutional neural network as our visual backbone. Following recent representation learning papers [18, 5, 8], we adopt a standard ResNet-50 [21] as our feature extractor throughout this paper to facilitate fair comparison. We remove the last linear classification layer and the preceding global average pooling layer to keep the spatial dimension. Thus, the output feature map from the visual backbone will have size $2048 \times R \times R$, where R is the output resolution (which is 1/32 of the input resolution).

Textual Backbone. The textual backbone encodes the input caption into a feature vector that captures the meaning of

each word token. In this paper, we adopt a Transformer [56] architecture as our textual backbone. Specially, we implement a 4-layer 1024-wide model with 16 self-attention heads. Similar to Desai *et al*. [8], we replace the activation function from ReLU to GELU [22] for its better empirical performance. We refer the readers to Vaswani *et al*. [56] for more architectural details. Before feeding the caption in, we first tokenize it into a lower-cased byte pair encoding (BPE) [50] with a vocabulary size of 10K. This results in almost no out-of-vocab unknown (`[UNK]`) tokens in our experiments. We also pad the input sequence with start of sequence (`[SOS]`) and end of sequence (`[EOS]`) tokens to mark the boundary. The output feature vector from the textual backbone has size $1024 \times L$ where L is the caption length after tokenization.

3.3. Contrastive Loss

Given a batch of feature pairs extracted from visual and textual backbones: $\{(\boldsymbol{x}_{\mathrm{V},k}, \boldsymbol{x}_{\mathrm{T},k}) \mid 1 \leq k \leq n\}$ (where n is the batch size), we transform each feature with a global average pooling and a single 1024-dimension fully-connected layer. The resulting visual and textual features are denoted $\boldsymbol{y}_{\mathrm{V},k}$ and $\boldsymbol{y}_{\mathrm{T},k}$ (both size 1024). Now, a straightforward way to guide the pre-training is to match $\boldsymbol{y}_{\mathrm{V},k}$ and $\boldsymbol{y}_{\mathrm{T},k}$ in the feature space using a simple L1/L2 regression loss. However, this will lead to a collapsed solution where all features are projected to the same location in the feature space [16].

Motivated by Chen *et al*. [5], we encourage the visual and textual backbones to not only project the features of *matching* image-caption pairs to be *closer* but also the features of *non-matching* pairs to be *further*. Concretely, there are n^2 image-caption pairs $\{(\boldsymbol{y}_{\mathrm{V},i}, \boldsymbol{y}_{\mathrm{T},j}) \mid 1 \leq i, j \leq n\}$ in total, among which only the n pairs with $i = j$ are positive (as they correspond to the same data) while the remaining $(n^2 - n)$ pairs are negative. We use the InfoNCE loss [35] to pull the positive pairs together and push the negative pairs apart to guide the pre-training (see Figure 2b):

$$\mathcal{L}_{\mathrm{C}} = -\sum_{i=1}^{n} \log \frac{\exp(\mathrm{sim}(\boldsymbol{y}_{\mathrm{V},i}, \boldsymbol{y}_{\mathrm{T},i})/\tau)}{\sum_{j \neq i} \exp(\mathrm{sim}(\boldsymbol{y}_{\mathrm{V},i}, \boldsymbol{y}_{\mathrm{T},j})/\tau)}, \quad (1)$$

where $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}/\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2$ is the cosine similarity between two vectors, and τ denotes a temperature parameter (which is set to 0.1 in our experiments).

Discussions. Contrastive learning is not the only way to bridge the vision and language modalities. It is also possible to use one modality as input and the other as output to form a supervised learning problem: *i.e*., image captioning [8] (image to caption) and image synthesis [45] (caption to image). However, the supervised formulation has a worse empirical performance than the contrastive one (see our comparisons with VirTex [8] in Table 4). Similar observations have also been made in our concurrent work [42]. We conjecture that this is because the relationship between image and caption is not one-to-one (*i.e*., a single image may be described in a multitude of ways, and vice versa). In this case, the encoding process (many-to-one projection) might be much easier than the decoding process (one-to-many projection).

3.4. Localization Loss

Applying the contrastive loss over the global visual and textual features (after average pooling) provides the model with a holistic sense of what objects are in the image. However, the model might not be able to correspond each instance with its spatial location. This greatly limits its effectiveness when transferred to localization-sensitive downstream tasks (*e.g*., object detection, instance segmentation). This is where the mouse trace can be helpful since it provides coarse localization information about the instances: *i.e*., where the annotators position their mouse when describing an object.

We provide an overview of our localization loss in Figure 2c. We first transform visual and textual features linearly using a 1024-dimension fully-connected layer. Note that we do not apply the global average pooling as we need to keep the spatial dimension to learn localization. Thus, the transformed visual feature $\boldsymbol{z}_{\mathrm{V},k}$ will have a size of $1024 \times R \times R$, and the transformed textual feature $\boldsymbol{z}_{\mathrm{V},k}$ will have a size of $1024 \times L$. We then compute the image-caption attention map as the normalized product between two feature maps:

$$\boldsymbol{\mathcal{M}}_k = \mathrm{softmax}(\boldsymbol{z}_{\mathrm{T,\,k}}^{\mathrm{T}} \times \boldsymbol{z}_{\mathrm{V,\,k}}), \quad (2)$$

which will then have the size of $L \times R \times R$. In $\boldsymbol{\mathcal{M}}_k$, each location (i, x, y) corresponds to (the probability of) whether the object described by the token i is located in the region of (x, y). We observe that this may be supervised using the mouse trace.

Given the fact that the mouse trace is synchronized with the caption, we first temporally crop the part of the mouse trace sequence that corresponds to each token in the caption. We then render the covered region of cropped mouse trace into a binary mask with resolution of R. Finally, we stack the rendered masks of all tokens together to generate the rendered attention $\hat{\boldsymbol{\mathcal{M}}}_k$. Since it has the same format and definition as the image-caption attention map $\boldsymbol{\mathcal{M}}_k$, we can use it to provide supervision on $\boldsymbol{\mathcal{M}}_k$ with a normalized L2 regression loss:

$$\mathcal{L}_{\mathrm{L}} = \sum_{k=1}^{n} \left\| \boldsymbol{\mathcal{M}}_k / \|\boldsymbol{\mathcal{M}}_k\|_2 - \hat{\boldsymbol{\mathcal{M}}}_k / \|\hat{\boldsymbol{\mathcal{M}}}_k\|_2 \right\|_2. \quad (3)$$

Discussions. The feature map from the visual backbone usually has a low resolution (*i.e*., $R = 7$ if the input size is 224×224), which largely limits the precision of the provided localized supervision. Therefore, we additionally apply the localization loss to the second last visual feature maps (which has 2× larger resolution) to provide supervision at a finer scale. The losses computed at different resolutions

are added together with equal weights. We note that using even higher resolutions than this leads to worse performance (see Table 5). A likely reason for this is that the mouse trace annotations from the datasets we use, and mouse traces in general, are intrinsically noisy. In this case, downsampling to a lower resolution removes some of the spurious correlations that otherwise might be introduced, at the cost of weaker overall supervision.

3.5. Implementation Details

Pre-Training Dataset. We use Localized Narratives [40] as our pre-training dataset as it provides large-scale localized textual annotations: *i.e.*, it annotates the whole COCO [29], Flickr30k [57], ADE20k [60], and part of Open Images [24] datasets with high-quality captions and synchronized mouse traces. In this paper, we present two variants of our LocTex: (1) a smaller one trained only with COCO images (which contains 118K images) to have a fair comparison with other "vision+language" baselines, and (2) a larger one trained on both COCO and Open Images data (which contains 809K annotated images) to test the scalability of our method. To compensate for the resolution difference with COCO, we downsample the images from Open Images by $0.6\times$.

Data Augmentation. We apply standard data augmentations for images: *i.e.*, random crop, random horizontal flip, color jittering and normalization. Following Desai *et al*. [8], we swap the 'left' and 'right' tokens in the caption when applying the horizontal flip. We limit the caption length to 60 tokens for computational efficiency: we pad the caption with zeros if its length is shorter than 60 or otherwise crop a random 60-token subsequence from the caption, which empirically helps to reduce overfitting.

Loss Functions. We assign $\mathcal{L}_{\text{C}}$ and $\mathcal{L}_{\text{L}}$ with equal weights as they are roughly of the same magnitude. The contrastive loss is computed locally at each GPU to save the communication bandwidth. This reduces the number of negative pairs, while empirically, the convergence rate is not affected.

Training Details. We pre-train the visual and textual backbones with a batch size of 1024 for 600 epochs. Optimization is carried out using stochastic gradient descent with a momentum of 0.9 and a weight decay of 10^{-4}. We use a learning rate of 0.4 for the visual backbone, 0.002 for the textual backbone, and 0.4 for the linear transforms. We adopt the cosine learning rate decay schedule [32] with a linear warmup for the first 20 epochs. We distribute the training over 8 NVIDIA V100 GPUs with synchronized batch normalization [38] and automatic mixed-precision [33] (from PyTorch [36]). The total training time is around 18 hours.

4. Experiments

In this section, we evaluate the effectiveness of our pre-trained visual backbone in various downstream vision tasks, including image classification, object detection and instance segmentation. The textual backbone also learns useful representations and can be transferred to language-related tasks in principle, though exploration of this is left as future work.

	# Pretrain Images	Annotations	mAP
Random Init	–	–	67.3
MoCo [18]	1.28M	self-supervised	79.4
PCL [28]	1.28M	self-supervised	83.1
SwAV [4]	1.28M	self-supervised	87.9
IN-Sup	1.28M	1 (1000-class) label	86.8
VirTex [8]	118K	1 caption	84.2
LocTex (Ours)	118K	1 localized caption	**88.4**
ICMLM [48]	118K	5 captions	87.5
VirTex [8]	118K	5 captions	88.7
LocTex (Ours)	809K	1 localized caption	**92.6**

Table 1: Results of linear classification on PASCAL VOC. Our LocTex outperforms supervised and self-supervised pre-training on ImageNet by **4-13%** while using around 60% of the annotated images. It also achieves **4%** higher accuracy than previous vision+language pre-training methods when trained with a similar amount of annotations.

4.1. Image Classification

Following the common protocol [18], we first evaluate our method by linear classification on frozen features: the pre-trained visual backbone is fixed and used to extract features.

Setup. We adopt the PASCAL VOC dataset [13] for our linear evaluation. We first resize all images to 224×224 and feed them into our pre-trained ResNet-50. We then apply global average pooling to extract 2048-dimensional image features. We train a separate SVM for each class on VOC07 `trainval` and report the mean AP (over 20 classes) on the `test` split. Following VirTex [8], we train multiple SVMs with different cost values from $\{0.01, 0.1, 1, 10\}$ and select the best SVM based on a 3-fold cross-validation.

Baselines. We compare our method with three sets of baselines: (1) ImageNet pre-training (IN-Sup) that pre-trains the model on the large-scale ImageNet dataset to perform image classification, (2) self-supervised learning [18, 28, 4] that pre-trains the model with a large number of unlabeled images, and (3) vision+language pre-training [48, 8] that pre-trains the model to perform image captioning on COCO.

Results. Training the classifier from scratch yields a rather poor performance because the size of PASCAL VOC is fairly small (with only 9K images). The widely-adopted ImageNet pre-training (IN-Sup) significantly boosts the accuracy; however, it requires massive annotations over a fine-grained class hierarchy. From Table 1, our LocTex achieves 1.6% higher accuracy than IN-Sup with only **10%** of annotated images,

	# Pretrain Images	10% Training Data						20% Training Data					
		AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Random Init	–	16.0	29.6	15.3	15.1	27.3	15.0	17.8	31.7	17.8	16.7	29.6	17.0
IN-Sup (10%)	128K	16.4	31.7	15.3	15.7	29.1	15.4	22.3	39.2	22.5	20.8	36.5	21.2
VirTex [8]	118K	23.7	41.9	24.0	21.5	38.6	21.4	28.9	47.4	30.6	25.6	44.1	26.2
LocTex (Ours)	118K	**25.0**	**43.2**	**25.7**	**22.4**	**39.8**	**22.4**	**29.8**	**48.9**	**31.1**	**26.4**	**45.2**	**27.2**
IN-Sup (50%)	640K	23.4	41.9	23.5	21.6	38.5	21.6	28.5	47.3	29.8	25.5	43.9	26.5
VirTex [8]	118K(×5)	26.3	44.1	27.1	23.4	40.9	23.8	30.7	49.4	32.3	27.1	45.9	27.9
LocTex (Ours)	809K	**27.3**	**45.8**	**28.2**	**24.2**	**42.1**	**24.9**	**31.8**	**50.9**	**33.8**	**27.8**	**47.3**	**28.9**
IN-Sup (100%)	1.28M	25.0	43.8	25.2	22.8	40.1	23.0	30.3	49.9	31.6	27.0	46.1	27.9

Table 2: Results of instance segmentation on COCO. Our LocTex consistently outperforms VirTex and IN-Sup under 10% and 20% data settings. We refer the readers to the appendix for detailed results under 50% and 100% data settings.

or **5.8%** higher with around 60% of annotated images. The superior performance comes from the use of cheap yet semantically dense localized caption annotations.

Previous vision+language pre-training methods [48, 8] were trained with five captions per image, which increases the annotation cost by 5×. To have a fair comparison, we compare our LocTex with the 1-caption VirTex [8]. With the same amount of pre-training images, our LocTex achieves more than **4%** higher accuracy, which is contributed by the better optimization formulation and the additional localization supervision (see Table 4). We are also on par in terms of the annotation cost as the extra mouse trace annotations we use can be acquired almost for free during the caption annotation [40]. We further scale our method up with the additional Open Images data. With a similar amount of annotated images, our LocTex outperforms the full VirTex by **4%** and ICMLM [48] by around **5%**.

4.2. Object Detection

We then evaluate our method by transferring our learned visual backbone to object detection. Here, the entire backbone is fine-tuned along with the object detector.

Setup. We adopt the PASCAL VOC dataset [13] for our detection evaluation. Different from the linear evaluation setup, we also include VOC12 `trainval` into the training set. For the object detector, we use Faster-RCNN [46] with ResNet-C4 backbone. Following He *et al*. [18], we add an extra batch normalization right after the visual backbone. We fine-tune all models for 24K iterations with linear warmup. The learning rate is initialized with 0.02 and decayed by 10× at 18K and 22K iteration. We distribute the training across 8 GPUs with a total batch size of 16.

Baselines. Apart from the full ImageNet pre-training baseline, we also scale it down with fewer pre-training images (10%, 20%, 50%) to match the annotation cost of VirTex and ours. We follow the same training protocol as `torchvision` and keep the number of epochs the same; otherwise, these models trained on smaller subsets are more prone to overfitting. These baselines are referred to as IN-Sup (k%).

	# Pretrain Images	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}
Random Init	–	33.8	60.2	33.1
IN-Sup (10%)	128K	42.6	72.0	43.8
MoCo [18]	118K	47.6	75.4	51.0
VirTex [8]	118K	51.7	79.6	56.5
LocTex (Ours)	118K	**53.9**	**80.9**	**59.8**
IN-Sup (50%)	640K	52.1	80.4	57.0
VirTex [8]	118K(×5)	55.3	81.3	61.0
LocTex (Ours)	809K	**56.9**	**82.4**	**63.2**
IN-Sup (100%)	1.28M	54.3	81.4	59.6

Table 3: Results of object detection on PASCAL VOC. Our LocTex surpasses VirTex and IN-Sup by **1.5-2.2%** and **4.8-11.3%** given a similar amount of pre-training images.

Results. We present our object detection results in Table 3. With a similar amount of pre-training images, our LocTex surpasses VirTex and IN-Sup by a large margin (**1.5-2.2%** and **4.8-11.3%**, respectively). Remarkably, LocTex matches the full ImageNet pre-training performance with more than **10**× fewer annotated images. The scaled-up version of LocTex further pushes the AP to 56.9%, which is 2.6% higher than the full ImageNet pre-training performance despite using **1.6**× fewer images.

4.3. Instance Segmentation

Finally, we evaluate our method on instance segmentation under the *limited data* setting. Similar to the detection setup, we train the visual backbone end-to-end with the model.

Setup. We use the COCO dataset [29] (with `train2017` and `val2017` split) for segmentation evaluation. We choose Mask R-CNN [20] with ResNet-C4 backbone as our model. We add the extra batch normalization to the visual backbone.

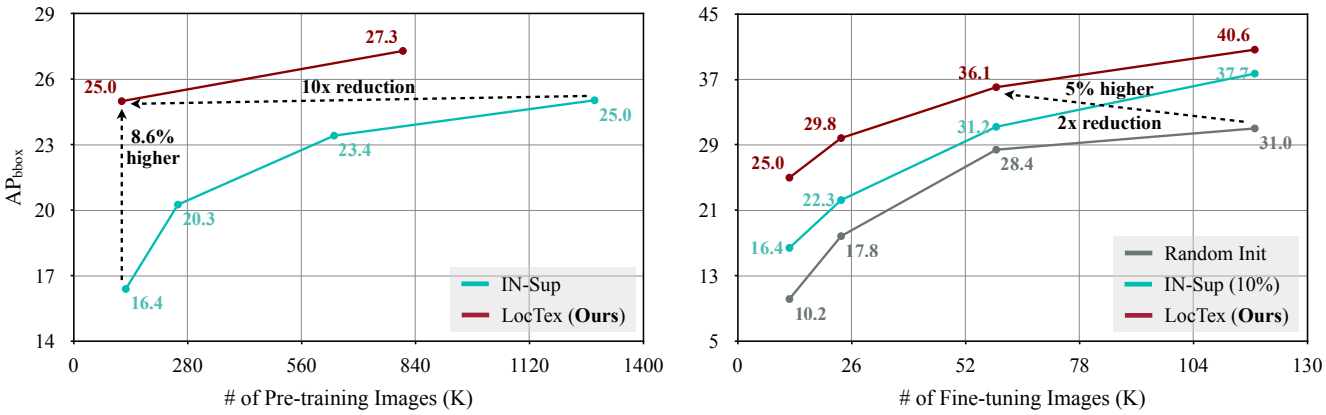


Figure 3: LocTex learns visual representations in a data-efficient manner: on COCO instance segmentation, it is able to reduce the pre-training dataset by **10**× without loss of accuracy or reduce the target dataset by **2**× with **5%** higher accuracy.

Following the 2× schedule, we train the model with 180K iterations. The learning rate is initialized with 0.02, multiplied by 0.01 at 120K and 160K iteration. As we target at the limited data setting, we sample a subset from COCO images (*e.g.*, 10%, 20%, 50%, 100%) for fine-tuning, and shrink the training schedule proportionally to the dataset size.

Results. From Table 2, our proposed LocTex consistently outperforms VirTex and IN-Sup under all data settings. We refer the readers to the appendix for detailed results under 50% and 100% data settings. In Figure 3, we further investigate our method from the data efficiency perspective:

– **Pre-Training Data.** Our LocTex can reduce the number of pre-training images by **10**× without loss of accuracy. With the same amount of pre-training data, it outperforms IN-Sup by more than **8%** in terms of AP. This translates into **2.4**× and **6.4**× lower annotation cost compared to pre-training with classification and segmentation labels. We refer the readers to the appendix for more details.

– **Fine-Tuning Data.** The end goal of a good pre-training is to reduce the amount of costly annotation in the target task. Our LocTex reduces the target dataset by **2**× while achieving more than **5%** higher accuracy than training from scratch. Under extremely limited data settings (*i.e.*, 5-10%), the improvement is even more significant: **2.7%** and **7.2%** AP boost compared with ImageNet pre-training and random initialization, with **2**× data reduction.

5. Analysis

In this section, we provide some additional analysis of our model to understand how it works and might be improved.

Effectiveness of $\mathcal{L}_C$ and $\mathcal{L}_L$. The two major components of LocTex are the formulation of contrastive learning ($\mathcal{L}_C$) and the use of low-cost mouse trace annotations ($\mathcal{L}_L$). Thus, we present some ablation analysis by removing one or both from our framework. VirTex [8] can be seen as our model

		VOC	COCO (10%)		COCO (20%)	
$\mathcal{L}_C$	$\mathcal{L}_L$	mAP	AP^{bbox}	AP^{mask}	AP^{bbox}	AP^{mask}
✓	✓	88.4	25.0	22.4	29.8	26.4
✓	✗	–0.9	–0.7	–0.6	–0.5	–0.5
✗	✗	–4.2	–1.3	–0.9	–0.9	–0.8

Table 4: The formulation of contrastive learning ($\mathcal{L}_C$) and the use of low-cost mouse trace annotations ($\mathcal{L}_L$) are important to the effectiveness of our visual representation learning.

removing both $\mathcal{L}_C$ and $\mathcal{L}_L$ (and using a predictive loss instead). From Table 4, both components contributes positively to our final performance on downstream vision tasks. We also observe that the contrastive loss is particularly effective on image classification; while the localization loss is more useful on instance segmentation. This phenomenon is well aligned with our design where $\mathcal{L}_C$ provides holistic semantic information and $\mathcal{L}_L$ offers detailed localization supervision.

Learned Image-Caption Attention Map. Although we focus on transferring the learned visual backbone to different downstream tasks, it is still fairly important to understand what the model actually learns from the pre-training stage. In Figure 4, we visualize the learned image-caption attention map. We refer the readers to the appendix for more examples. Here, the visualized attention maps are predicted from the second last visual feature map (with resolution of 14×14). We resize the attention maps to 224×224 and then overlay them to images. As shown in Figure 4, the learned attention maps have fairly accurate localization and are able to capture occluded and distant instances (*e.g.*, cars and buildings in the third example). This explains why our model transfers well to detection and segmentation. As the model is trained with open-vocabulary textual annotations, it is able to learn rich visual concepts, some of which (*e.g.*, helmets and goggles) are not even covered in the COCO categories. This shows great potential in the fine-grained localization tasks (such as

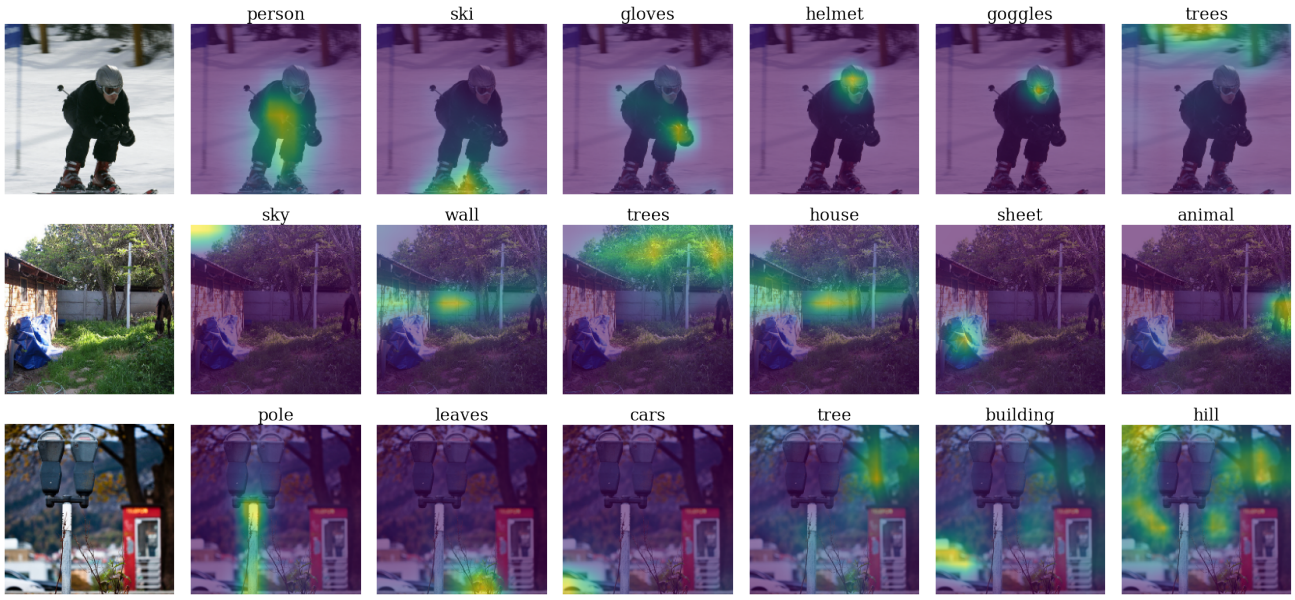


Figure 4: Visualization of learned image-caption attention maps (on COCO `val2017`). Our LocTex learns rich visual concepts (*e.g.*, helmets, goggles) and fairly accurate localization. We refer the readers to the appendix for further examples.

	VOC	COCO (10%)		COCO (20%)	
Supervision	mAP	AP^{bbox}	AP^{mask}	AP^{bbox}	AP^{mask}
1×	87.7	24.3	21.9	29.3	25.9
1×, 2×	88.4	25.0	22.4	29.8	26.4
1×, 2×, 4×	88.2	24.9	22.3	29.6	26.1
Oracle	**90.8**	**26.1**	**23.3**	**30.7**	**21.7**

Table 5: Analysis of mouse trace supervision. (1) Applying supervision at too low or too high resolution does not work well. (2) With oracle supervision, the performance is further boosted by **2%** on classification and **1%** on segmentation.

LVIS [17]), which is left as future work. Another interesting direction is to study the zero-shot transfer performance to detection/segmentation based on the learned attention maps.

Resolution of Mouse Trace Supervision. We explore different resolutions for mouse trace supervision. As shown in Table 5, applying supervision at both 1× and 2× resolutions works the best across different downstream tasks. 1× alone does not work well due to its low resolution (7×7) while 4× introduces too much noise from the mouse trace annotation.

Performance "Upper Bound". We further investigate the performance upper bound of our method given perfect mouse trace annotations. We synthesize the clean image-caption attention maps using ground-truth COCO segmentation masks. Specifically, we first match each token in the caption with the COCO category names (as well as their synonyms and parent classes). For each token with a match, we compute the intersection-over-union (IoU) between its corresponding mouse trace and every instance mask in the matching category. Finally, we aggregate these instance masks with high IoUs as our oracle image-caption attention maps. The IoU matching process helps to deal with the case where the token in the caption only refers to one of the multiple instances from the category. Note that we apply the oracle supervision still at 1× and 2× scale to mimic the coarse resolution of real mouse traces. In Table 5, our LocTex trained with oracle supervision further pushes the performance by **2%** on the PASCAL VOC image classification and **1%** on the COCO instance segmentation.

Training Efficiency. In addition to annotation efficiency, our LocTex pre-training is also very efficient in computation. Its training cost is comparable with ImageNet supervised pre-training. We refer the readers to the appendix for details.

6. Conclusion

In this paper, we introduce LocTex to reduce the practical costs of data annotation by taking advantage of low-cost, multi-modal labels including free-form captions and mouse-over gestures. We adopt a cross-modal contrastive pre-training approach using images and captions, and propose to supervise the image-caption attention map via rendered mouse traces to provide coarse localization information. Extensive experiments verify that the visual features learned through our approach can be effectively and efficiently transferred to downstream tasks including image classification, object detection, and instance segmentation. We hope that our approach will provide a simple but strong baseline and inspire future exploration into how to extract more value from rich yet noisy localized textual annotations.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. In *CVPR*, 2018.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners. In *NeurIPS*, 2020.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020.

[6] Andrew M Dai and Quoc V Le. Semi-supervised Sequence Learning. In *NeurIPS*, 2015.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[8] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. In *CVPR*, 2021.

[9] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *ICCV*, 2015.

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *ICML*, 2014.

[11] Jeff Donahue and Karen Simonyan. Large Scale Adversarial Representation Learning. In *NeurIPS*, 2019.

[12] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially Learned Inference. In *ICLR*, 2017.

[13] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge: A Retrospective. *IJCV*, 2015.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. In *NeurIPS*, 2020.

[17] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *CVPR*, 2019.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020.

[19] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking ImageNet Pre-training. In *ICCV*, 2019.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[22] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv*, 2016.

[23] Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning Visual Features from Large Weakly Supervised Data. In *ECCV*, 2016.

[24] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *IJCV*, 2020.

[25] Quoc V Le. Building High-Level Features using Large Scale Unsupervised Learning. In *ICASSP*, 2013.

[26] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *ICML*, 2009.

[27] Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten. Learning Visual N-Grams from Web Data. In *ICCV*, 2017.

[28] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical Contrastive Learning of Unsupervised Representations. In *ICLR*, 2021.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[30] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast Interactive Object Annotation with Curve-GCN. In *CVPR*, 2019.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015.

[32] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *ICLR*, 2017.

[33] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Wu Hao. Mixed Precision Training. In *ICLR*, 2018.

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *ECCV*, 2016.

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv*, 2018.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019.

[37] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context Encoders: Feature Learning by Inpainting. In *CVPR*, 2016.

[38] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. MegDet: A Large Mini-Batch Object Detector. In *CVPR*, 2018.

[39] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *NAACL*, 2018.

[40] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting Vision and Language with Localized Narratives. In *ECCV*, 2020.

[41] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Learning visual representations using images with captions. In *CVPR*, 2007.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv*, 2021.

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 2020.

[44] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding Transfer Learning for Medical Imaging. In *NeurIPS*, 2019.

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv*, 2021.

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[48] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning Visual Representations with Caption Annotations. In *ECCV*, 2020.

[49] Ozan Sener and Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *ICLR*, 2018.

[50] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *ACL*, 2016.

[51] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Object Detection from Scratch with Deep Supervision. *TPAMI*, 2019.

[52] Nitish Srivastava, Ruslan Salakhutdinov, et al. Multimodal Learning with Deep Boltzmann Machines. In *NeurIPS*, 2012.

[53] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *ICCV*, 2017.

[54] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The New Data in Multimedia Research. *CACM*, 2016.

[55] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. In *ECCV*, 2019.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017.

[57] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. *TACL*, 2014.

[58] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *ECCV*, 2016.

[59] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive Learning of Medical Visual Representations from Paired Images and Text. *arXiv*, 2020.

[60] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Understanding of Scenes through the ADE20K Dataset. *IJCV*, 2019.

[61] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking Pre-training and Self-training. In *NeurIPS*, 2020.