ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision

Wonjae Kim [*1 †] **Bokyung Son** [*1] **Ildoo Kim** [2]

Abstract

Vision-and-Language Pre-training (VLP) has improved performance on various joint vision-and-language downstream tasks. Current approaches to VLP heavily rely on image feature extraction processes, most of which involve region supervision (e.g., object detection) and the convolutional architecture (e.g., ResNet). Although disregarded in the literature, we find it problematic in terms of both (1) efficiency/speed, that simply extracting input features requires much more computation than the multimodal interaction steps; and (2) expressive power, as it is upper bounded to the expressive power of the visual embedder and its predefined visual vocabulary. In this paper, we present a minimal VLP model, Vision-and-Language Transformer (ViLT), monolithic in the sense that the processing of visual inputs is drastically simplified to just the same convolution-free manner that we process textual inputs. We show that ViLT is up to tens of times faster than previous VLP models, yet with competitive or better downstream task performance. Our code and pre-trained weights are available at https://github.com/dandelin/vilt.

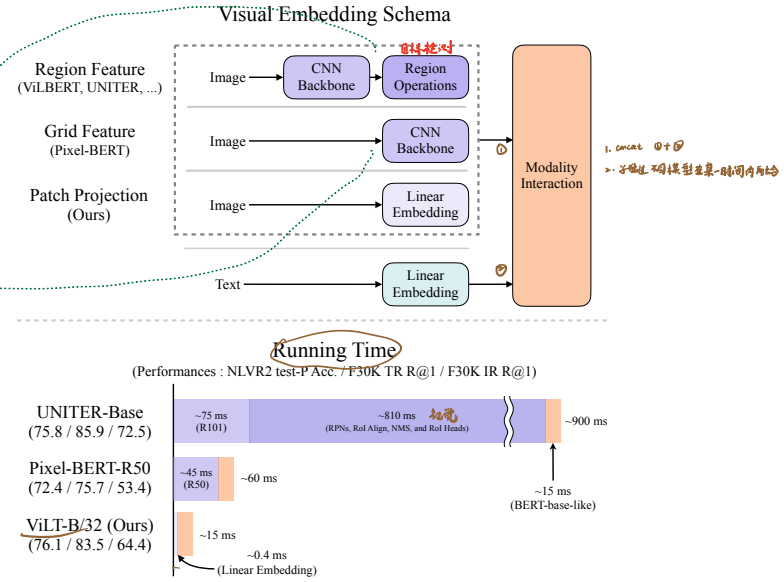

Figure 1. Visual comparison of conventional VLP architectures and our proposed ViLT. We have entirely removed convolutional neural networks from the VLP pipeline without hurting performance on downstream tasks. ViLT is the first VLP model of which the modal-specific components require *less* computation than the transformer component for multimodal interactions.

1. Introduction

The pre-train-and-fine-tune scheme has been expanded to a joint domain of vision and language, giving birth to the category of *Vision-and-Language Pre-training (VLP)* models (Lu et al., 2019; Chen et al., 2019; Su et al., 2019; Li et al., 2019; Tan & Bansal, 2019; Li et al., 2020a; Lu et al., 2020; Cho et al., 2020; Qi et al., 2020; Zhou et al., 2020; Huang et al., 2020; Li et al., 2020b; Gan et al., 2020; Yu et al., 2020; Zhang et al., 2021). These models are pre-trained with image text matching and masked language modeling objectives[1] on images and their aligned descriptions, and are fine-tuned on vision-and-language downstream tasks where the inputs involve two modalities.

To be fed into VLP models, image pixels need to be initially embedded in a dense form alongside language tokens. Since the seminal work of Krizhevsky et al. (2012), deep convolutional networks have been regarded as essential for this visual embedding step. Most VLP models employ an object detector pre-trained on the Visual Genome dataset (Krishna et al., 2017) annotated with 1,600 object classes and 400 attribute classes as in Anderson et al. (2018). Pixel-

[*]Equal contribution [†]Current affiliation: NAVER AI Lab, Seongnam, Gyeonggi, Republic of Korea. [1]Kakao Enterprise, Seongnam, Gyeonggi, Republic of Korea [2]Kakao Brain, Seongnam, Gyeonggi, Republic of Korea. Correspondence to: Wonjae Kim <wonjae.kim@navercorp.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]While some works employ additional objectives and data structures, these two objectives apply to almost every VLP model.

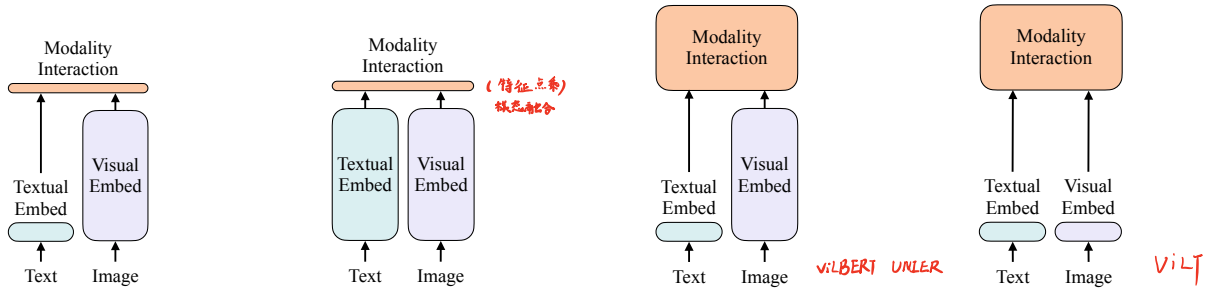

Figure 2. Four categories of vision-and-language models. The height of each rectangle denotes its relative computational size. VE, TE, and MI are short for visual embedder, textual embedder, and modality interaction, respectively.

BERT (Huang et al., 2020) is one exception of this trend, as it uses ResNet variants (He et al., 2016; Xie et al., 2017) pre-trained on ImageNet classification (Russakovsky et al., 2015) embedding pixels in place of object detection modules.

To this date, most VLP studies have focused on improving performance by increasing the power of visual embedders. The shortcomings of having a heavy visual embedder are often disregarded in academic experiments because region features are commonly cached in advance at training time to ease the burden of feature extraction. However, the limitations are still evident in real-world applications as the queries in the wild have to undergo a slow extraction process.

To this end, we shift our attention to the lightweight and fast embedding of visual inputs. Recent work (Dosovitskiy et al., 2020; Touvron et al., 2020) demonstrated that using a simple linear projection of a patch is effective enough to embed pixels before feeding them into transformers. Whereas being the solid mainstream for text (Devlin et al., 2019), it is only recently that transformers (Vaswani et al., 2017) are used for images as well. We presume that the transformer module– used for modality interaction in VLP models– can also manage to process visual features in place of a convolutional visual embedder, just as it processes textual features.

This paper proposes the Vision-and-Language Transformer (ViLT) that handles two modalities in a single unified manner. It mainly differs from previous VLP models in its shallow, convolution-free embedding of pixel-level inputs. Removing deep embedders solely dedicated to visual inputs significantly cuts down the model size and running time by design. Figure 1 shows that our parameter-efficient model is tens of times faster than VLP models with region features and at least four times faster than those with grid features while exhibiting similar or even better performance on vision-and-language downstream tasks.

Our key contributions can be summarized as follows:

- ViLT is the *simplest* architecture by far for a vision-and-language model as it commissions the transformer module to extract and process visual features in place of a separate deep visual embedder. This design inherently leads to significant runtime and parameter efficiency.
- For the first time, we achieve competent performance on vision-and-language tasks without using region features or deep convolutional visual embedders in general.
- Also, for the first time, we empirically show that whole word masking and image augmentations that were unprecedented in VLP training schemes further drive downstream performance.

2. Background

2.1. Taxonomy of Vision-and-Language Models

We propose a taxonomy of vision-and-language models based on two points: (1) whether the two modalities have an even level of expressiveness in terms of dedicated parameters and/or computation; and (2) whether the two modalities interact in a deep network. A combination of these points leads to four archetypes in Figure 2.

The *visual semantic embedding* (VSE) models such as VSE++ (Faghri et al., 2017) and SCAN (Lee et al., 2018) belong to Figure 2a. They use separate embedders for image and text, with the former being much heavier. Then, they represent the similarity of the embedded features from the two modalities with simple dot products or shallow attention layers.

CLIP (Radford et al., 2021) belongs to Figure 2b as it uses separate but equally expensive transformer embedders for each modality. Interaction between the pooled image vector and text vector is still shallow (dot product). Despite CLIP's remarkable zero-shot performance on image-to-text

retrieval, we could not observe the same level of performance on other vision-and-language downstream tasks. For instance, fine-tuning the MLP head on NLVR2 (Suhr et al., 2018) with the dot product of pooled visual and textual vectors from CLIP as the multimodal representation gives a low dev accuracy of 50.99 ± 0.38 (ran with three different seeds); as chance level accuracy is 0.5, we conclude that the representations are incapable of learning this task. It also matches the findings of Suhr et al. (2018) that all models with simply fused multimodal representation failed to learn NLVR2.

This result backs up our speculation that simple fusion of outputs even from high-performing unimodal embedders may not be sufficient to learn complex vision-and-language tasks, bolstering the need for a more rigorous inter-modal interaction scheme.

Unlike models with shallow interaction, the more recent VLP models that fall under Figure 2c use a deep transformer to model the interaction of image and text features. Aside from the interaction module, however, convolutional networks are still involved in extracting and embedding image features, which accounts for most of the computation as depicted in Figure 1. Modulation-based vision-and-language models (Perez et al., 2018; Nguyen et al., 2020) also fall under Figure 2c, with their visual CNN stems corresponding to visual embedder, RNNs producing the modulation parameters to textual embedder, and modulated CNNs to modality interaction.

Our proposed ViLT is the first model of type Figure 2d where the embedding layers of raw pixels are shallow and computationally light as of text tokens. This architecture thereby concentrates most of the computation on modeling modality interactions.

2.2. Modality Interaction Schema

At the very core of contemporary VLP models lie transformers. They get visual and textual embedding sequences as input, model inter-modal and optionally intra-modal interactions throughout layers, then output a contextualized feature sequence.

Bugliarello et al. (2020) classifies interaction schema into two categories: (1) *single-stream* approaches (e.g., VisualBERT (Li et al., 2019), UNITER (Chen et al., 2019)) where layers collectively operate on a concatenation of image and text inputs; and (2) *dual-stream* approaches (e.g., ViLBERT (Lu et al., 2019), LXMERT (Tan & Bansal, 2019)) where the two modalities are not concatenated at the input level. We follow the single-stream approach for our interaction transformer module because the dual-stream approach introduces additional parameters.

2.3. Visual Embedding Schema

Whereas all performant VLP models share the same textual embedder– tokenizer from pre-trained BERT, word and position embeddings resembling those of BERT– they differ on visual embedders. Still, in most (if not all) cases, visual embedding is the bottleneck of existing VLP models. We focus on cutting corners on this step by introducing patch projection instead of using region or grid features for which heavy extraction modules are used.

Region Feature. VLP models dominantly utilize region features, also known as bottom-up features (Anderson et al., 2018). They are obtained from an off-the-shelf object detector like Faster R-CNN (Ren et al., 2016).

The general pipeline of generating region features is as follows. First, a region proposal network (RPN) proposes regions of interest (RoI) based on the grid features pooled from the CNN backbone. Non-maximum suppression (NMS) then reduces the number of RoIs to a few thousand. After being pooled by operations such as RoI Align (He et al., 2017), the RoIs go through RoI heads and become region features. NMS is again applied to every class, finally reducing the number of features under a hundred.

The above process involves several factors that affect the performance and runtime: the backbone, the style of NMS, the RoI heads. Previous works were lenient with controlling these factors, making varying choices from each other as listed in Table 7.[2]

- Backbone: ResNet-101 (Lu et al., 2019; Tan & Bansal, 2019; Su et al., 2019) and ResNext-152 (Li et al., 2019; 2020a; Zhang et al., 2021) are two commonly used backbones.
- NMS: NMS is typically done in a *per-class* fashion. Applying NMS to each and every class becomes a major runtime bottleneck with a large number of classes, *e.g.* 1.6K in the VG dataset (Jiang et al., 2020). *Class-agnostic* NMS was recently introduced to tackle this issue (Zhang et al., 2021).
- RoI head: C4 heads were initially used (Anderson et al., 2018). FPN-MLP heads were introduced later (Jiang et al., 2018). As heads operate for each and every RoI, they pose a substantial runtime burden.

However lightweight, object detectors are less likely to be faster than the backbone or a single-layer convolution. Freezing the visual backbone and caching the region features in advance only helps at training time and not during

[2] Bugliarello et al. (2020) showed that a controlled setup bridges the performance gap of various region-feature-based VLP models.

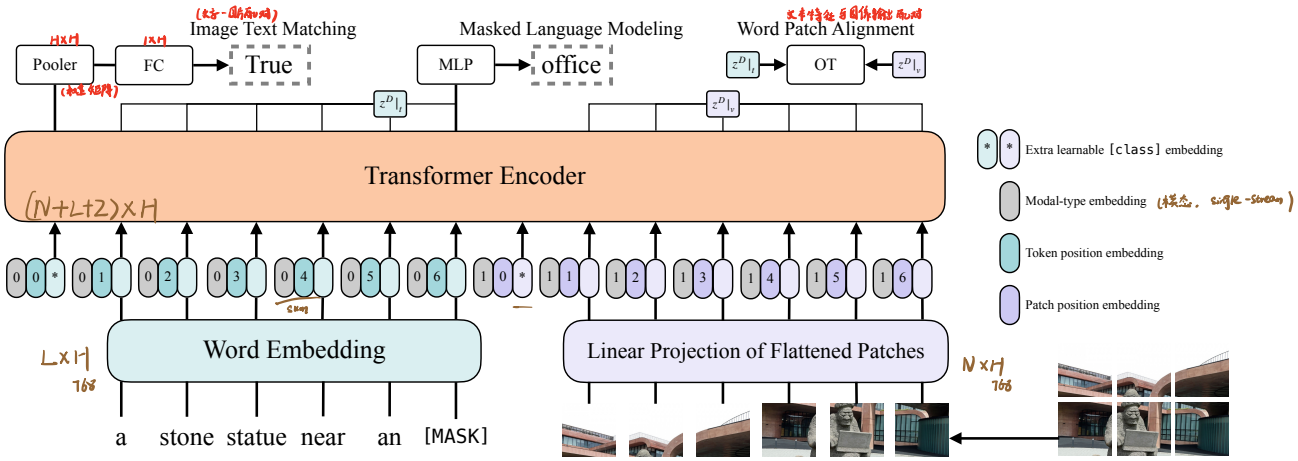

Figure 3. Model overview. Illustration inspired by Dosovitskiy et al. (2020).

inference, not to mention that it could hold performance back.

Grid Feature. Besides detector heads, the output feature grid of convolutional neural networks such as ResNets can also be used as visual features for vision-and-language pre-training. Direct use of grid features was first proposed by VQA-specific models (Jiang et al., 2020; Nguyen et al., 2020), mainly to avoid using severely slow region selection operations.

X-LXMERT (Cho et al., 2020) revisited grid features by fixing the region proposals to grids instead of those from the region proposal networks. However, their caching of features excluded further tuning of the backbone.

Pixel-BERT is the only VLP model that replaces the VG-pre-trained object detector with a ResNet variant backbone pre-trained with ImageNet classification. Unlike frozen detectors in region-feature-based VLP models, the backbone of Pixel-BERT is tuned during vision-and-language pre-training. The downstream performance of Pixel-BERT with ResNet-50 falls below region-feature-based VLP models, but it matches that of other competitors with the use of a much heavier ResNeXt-152.

We claim that grid features are not the go-to option, however, since deep CNNs are still expensive that they account for a large portion of the whole computation as in Figure 1.

Patch Projection. To minimize overhead, we adopt the simplest visual embedding scheme: *linear projection* that operates on image patches. The patch projection embedding was introduced by ViT (Dosovitskiy et al., 2020) for image classification tasks. Patch projection drastically simplifies the visual embedding step to the level of textual embedding, which also consists of simple projection (lookup) operations. We use a 32×32 patch projection which only requires 2.4M parameters. This is in sharp contrast to complex ResNe(X)t backbones[3] and detection components. Its running time is also ignorable as shown in Figure 1. We make a detailed runtime analysis in Section 4.6.

3. Vision-and-Language Transformer

3.1. Model Overview

ViLT has a succinct architecture as a VLP model with a minimal visual embedding pipeline and following the single-stream approach.

We deviate from the literature that we initialize the interaction transformer weights from pre-trained ViT instead of BERT. Such initialization exploits the power of the interaction layers to process visual features while lacking a separate deep visual embedder. [4]

$$\bar{t} = [t_{\text{class}}; t_1 T; \cdots; t_L T] + T^{\text{pos}} \quad (1)$$

$$\bar{v} = [v_{\text{class}}; v_1 V; \cdots; v_N V] + V^{\text{pos}} \quad (2)$$

$$z^0 = [\bar{t} + t^{\text{type}}; \bar{v} + v^{\text{type}}] \quad (3)$$

$$\hat{z}^d = \text{MSA}(\text{LN}(z^{d-1})) + z^{d-1}, \qquad d = 1 \dots D \quad (4)$$

$$z^d = \text{MLP}(\text{LN}(\hat{z}^d)) + \hat{z}^d, \qquad d = 1 \dots D \quad (5)$$

$$p = \tanh(z_0^D W_{\text{pool}}) \quad (6)$$

ViT consists of stacked blocks that include a multiheaded self-attention (MSA) layer and an MLP layer. The position of layer normalization (LN) in ViT is the only difference from BERT: LN comes after MSA and MLP in BERT ("post-norm") and before in ViT ("pre-norm"). The input

[3] Parameters for R50 is 25M, R101 is 44M, and X152 is 60M.

[4] We also experimented with initializing the layers from BERT weights and using the pre-trained patch projection from ViT, but it did not work.

text $t \in \mathbb{R}^{L\times|V|}$ is embedded to $\bar{t} \in \mathbb{R}^{L\times H}$ with a word embedding matrix $T \in \mathbb{R}^{|V|\times H}$ and a position embedding matrix $T^{\text{pos}} \in \mathbb{R}^{(L+1)\times H}$.

The input image $I \in \mathbb{R}^{C\times H\times W}$ is sliced into patches and flattened to $v \in \mathbb{R}^{N\times(P^2\cdot C)}$ where (P, P) is the patch resolution and $N = HW/P^2$. Followed by linear projection $V \in \mathbb{R}^{(P^2\cdot C)\times H}$ and position embedding $V^{\text{pos}} \in \mathbb{R}^{(N+1)\times H}$, v is embedded into $\bar{v} \in \mathbb{R}^{N\times H}$.

The text and image embeddings are summed with their corresponding modal-type embedding vectors $t^{\text{type}}, v^{\text{type}} \in \mathbb{R}^H$, then are concatenated into a combined sequence z^0. The contextualized vector z is iteratively updated through D-depth transformer layers up until the final contextualized sequence z^D. p is a pooled representation of the whole multimodal input, and is obtained by applying linear projection $W_{\text{pool}} \in \mathbb{R}^{H\times H}$ and hyperbolic tangent upon the first index of sequence z^D.

For all experiments, we use weights from ViT-B/32 pre-trained on ImageNet, hence the name ViLT-B/32.[5] Hidden size H is 768, layer depth D is 12, patch size P is 32, MLP size is 3,072, and the number of attention heads is 12.

3.2. Pre-training Objectives

We train ViLT with two objectives commonly used to train VLP models: image text matching (ITM) and masked language modeling (MLM).

Image Text Matching. We randomly replace the aligned image with a different image with the probability of 0.5. A single linear layer ITM head projects the pooled output feature p to logits over binary class, and we compute negative log-likelihood loss as our ITM loss.

Plus, inspired by the word region alignment objective in Chen et al. (2019), we design word patch alignment (WPA) that computes the alignment score between two subsets of z^D: $z^D|_t$ (textual subset) and $z^D|_v$ (visual subset), using the inexact proximal point method for optimal transports (IPOT) (Xie et al., 2020). We set the hyperparameters of IPOT following Chen et al. (2019) ($\beta = 0.5, N = 50$), and add the approximate wasserstein distance multiplied by 0.1 to the ITM loss.

Masked Language Modeling. This objective is to predict the ground truth labels of masked text tokens t_{masked} from its contextualized vector $z^D_{\text{masked}}|_t$. Following the heuristics of Devlin et al. (2019), we randomly mask t with the probability of 0.15.

[5] ViT-B/32 is pre-trained with ImageNet-21K and fine-tuned on ImageNet-1K for image classification. We expect that weights pre-trained on larger datasets (e.g., JFT-300M) would yield better performance.

We use a two-layer MLP MLM head that inputs $z^D_{\text{masked}}|_t$ and outputs logits over vocabulary, just as the MLM objective of BERT. The MLM loss is then computed as the negative log-likelihood loss for the masked tokens.

3.3. Whole Word Masking

Whole word masking is a masking technique that masks all consecutive subword tokens that compose a whole word. It is shown to be effective on downstream tasks when applied to original and Chinese BERT (Cui et al., 2019).

We hypothesize that whole word masking is particularly crucial for VLP in order to make full use of information from the other modality. For example, the word "giraffe" is tokenized into three wordpiece tokens `["gi", "##raf", "##fe"]` with the pre-trained `bert-base-uncased` tokenizer. If not all tokens are masked, say, `["gi", "[MASK]", "##fe"]`, the model may solely rely on the nearby two language tokens `["gi", "##fe"]` to predict the masked `"##raf"` rather than using the information from the image.

We mask whole words with a mask probability of 0.15 during pre-training. We discuss its impact in Section 4.5.

3.4. Image Augmentation

Image augmentation reportedly improves the generalization power of vision models (Shorten & Khoshgoftaar, 2019). DeiT (Touvron et al., 2020) that builds on ViT experimented with various augmentation techniques (Zhang et al., 2017; Yun et al., 2019; Berman et al., 2019; Hoffer et al., 2020; Cubuk et al., 2020), and found them beneficial for ViT training. However, the effects of image augmentation have not been explored within VLP models. Caching visual features restrains region-feature-based VLP models from using image augmentation. Notwithstanding its applicability, neither did Pixel-BERT study its effects.

To this end, we apply RandAugment (Cubuk et al., 2020) during fine-tuning. We use all the original policies except two: color inversion, because texts often contain color information as well, and cutout, as it may clear out small but important objects dispersed throughout the whole image. We use $N = 2, M = 9$ as the hyperparameters. We discuss its impact in Section 4.5 and Section 5.

4. Experiments

4.1. Overview

We use four datasets for pre-training: Microsoft COCO (MSCOCO) (Lin et al., 2014), Visual Genome (VG) (Krishna et al., 2017), SBU Captions (SBU) (Ordonez et al., 2011), and Google Conceptual Captions (GCC) (Sharma

Table 1. Pre-training dataset statistics. Caption length is the length of tokens from pre-trained `bert-base-uncased` tokenizer. † GCC and SBU provide only image urls, so we collect the images from urls which were still accessible.

Dataset	# Images	# Captions	Caption Length
MSCOCO	113K	567K	11.81 ± 2.81
VG	108K	5.41M	5.53 ± 1.76
GCC†	3.01M	3.01M	10.66 ± 4.93
SBU†	867K	867K	15.0 ± 7.74

et al., 2018). Table 1 reports the dataset statistics.

We evaluate ViLT on two widely explored types of vision-and-language downstream tasks: for *classification*, we use VQAv2 (Goyal et al., 2017) and NLVR2 (Suhr et al., 2018), and for *retrieval*, we use MSCOCO and Flickr30K (F30K) (Plummer et al., 2015) re-splited by Karpathy & Fei-Fei (2015). For the classification tasks, we fine-tune three times with different initialization seeds for the head and data ordering and report the mean scores. We report the standard deviation in Table 5 along with ablation studies. For the retrieval tasks, we only fine-tune once.

4.2. Implementation Details

For all experiments, we use AdamW optimizer (Loshchilov & Hutter, 2018) with base learning rate of 10^{-4} and weight decay of 10^{-2}. The learning rate was warmed up for 10% of the total training steps and was decayed linearly to zero for the rest of the training. Note that downstream performance may be further improved if we customize the hyperparameters to each task.

We resize the shorter edge of input images to 384 and limit the longer edge to under 640 while preserving the aspect ratio. This resizing scheme is also used during object detection in other VLP models, but with a larger size of the shorter edge (800). Patch projection of ViLT-B/32 yields $12 \times 20 = 240$ patches for an image with a resolution of 384×640. As this is a rarely reached upper limit, we sample 200 patches at maximum during pre-training. We interpolate V^{pos} of ViT-B/32 to fit the size of each image and pad the patches for batch training. Note that the resulting image resolution is four times smaller than $800 \times 1{,}333$, which is the size that all other VLP models use for inputs to their visual embedders.

We use the `bert-base-uncased` tokenizer to tokenize text inputs. Instead of fine-tuning from pre-trained BERT, we learn the textual embedding-related parameters t_{class}, T, and T^{pos} from scratch. Although beneficial *prima facie*, employing a pre-trained text-only BERT does not guarantee performance gain for vision and language downstream tasks. Counterevidence has already been reported by Tan & Bansal (2019), where initializing with pre-trained BERT parameters led to weaker performance than pre-training from scratch.

Table 2. Comparison of ViLT-B/32 with other models on downstream classification tasks. We use MCAN (Yu et al., 2019) and MaxEnt (Suhr et al., 2018) for VQAv2 and NLVR2 w/o VLP SOTA results. † additionally used GQA, VQAv2, VG-QA for pre-training. ‡ made additional use of the Open Images (Kuznetsova et al., 2020) dataset. ⓐ indicates RandAugment is applied during fine-tuning. ⊕ indicates model trained for a longer 200K pre-training steps.

Visual Embed	Model	Time (ms)	VQAv2 test-dev	NLVR2 dev	NLVR2 test-P
Region	w/o VLP SOTA	~900	70.63	54.80	53.50
	ViLBERT	~920	70.55	-	-
	VisualBERT	~925	70.80	67.40	67.00
	LXMERT	~900	72.42	74.90	74.50
	UNITER-Base	~900	72.70	75.85	75.80
	OSCAR-Base†	~900	73.16	78.07	78.36
	VinVL-Base†‡	~650	75.95	82.05	83.08
Grid	Pixel-BERT-X152	~160	74.45	76.50	77.20
	Pixel-BERT-R50	~60	71.35	71.70	72.40
Linear	ViLT-B/32	~15	70.33	74.41	74.57
	ViLT-B/32ⓐ	~15	70.85	74.91	75.57
	ViLT-B/32ⓐ⊕	~15	71.26	75.70	76.13

We pre-train ViLT-B/32 for 100K or 200K steps on 64 NVIDIA V100 GPUs with a batch size of 4,096. For all downstream tasks, we train for ten epochs with a batch size of 256 for VQAv2/retrieval tasks and 128 for NLVR2.

4.3. Classification Tasks

We evaluate ViLT-B/32 on two commonly used datasets: VQAv2 and NLVR2. We use a two-layer MLP of hidden size 1,536 as the fine-tuned downstream head.

Visual Question Answering. The VQAv2 task asks for answers given pairs of an image and a question in natural language. The annotated answers are originally in free-form natural language, but it is a common practice to convert the task to a classification task with 3,129 answer classes. Following this practice, we fine-tune ViLT-B/32 on the VQAv2 train and validation sets while reserving 1,000 validation images and their related questions for internal validation.

We report the test-dev score results[6] from the submission to the evaluation server. ViLT falls short of VQA score compared to other VLP models with a heavy visual embedder. We suspect a detached object representation generated by the object detector eases the training of VQA since questions in VQA typically ask about objects.

[6] VQA score is calculated by comparing the inferred answer to 10 ground-truth answers: see `https://visualqa.org/evaluation.html` for details.

Table 3. Comparison of ViLT-B/32 with other VLP models on downstream zero-shot retrieval tasks. We exclude the models of which zero-shot retrieval performances were not reported in their original papers. † is pre-trained with a 10M proprietary vision-and-language dataset in addition to the 4M dataset of GCC+SBU. ⊕ indicates model trained for a longer 200K pre-training steps.

Visual Embed	Model	Time (ms)	Zero-Shot Text Retrieval						Zero-Shot Image Retrieval					
			Flickr30k (1K)			MSCOCO (5K)			Flickr30k (1K)			MSCOCO (5K)		
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Region	ViLBERT	~900	-	-	-	-	-	-	31.9	61.1	72.8	-	-	-
	Unicoder-VL	~925	64.3	85.8	92.3	-	-	-	48.4	76.0	85.2	-	-	-
	UNITER-Base	~900	80.7	95.7	98.0	-	-	-	66.2	88.4	92.9	-	-	-
	ImageBERT†	~925	70.7	90.2	94.0	44.0	71.2	80.4	54.3	79.6	87.5	32.3	59.0	70.2
Linear	ViLT-B/32	~15	69.7	91.0	96.0	53.4	80.7	88.8	51.3	79.9	87.9	37.3	67.4	79.0
	ViLT-B/32⊕	~15	73.2	93.6	96.5	56.5	82.6	89.6	55.0	82.5	89.8	40.4	70.0	81.1

Table 4. Comparison of ViLT-B/32 with other models on downstream retrieval tasks. We use SCAN for w/o VLP SOTA results. † additionally used GQA, VQAv2, VG-QA for pre-training. ‡ additionally used the Open Images dataset. ⓐ indicates RandAugment is applied during fine-tuning. ⊕ indicates model trained for a longer 200K pre-training steps.

Visual Embed	Model	Time (ms)	Text Retrieval						Image Retrieval					
			Flickr30k (1K)			MSCOCO (5K)			Flickr30k (1K)			MSCOCO (5K)		
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Region	w/o VLP SOTA	~900	67.4	90.3	95.8	50.4	82.2	90.0	48.6	77.7	85.2	38.6	69.3	80.4
	ViLBERT-Base	~920	-	-	-	-	-	-	58.2	84.9	91.5	-	-	-
	Unicoder-VL	~925	86.2	96.3	99.0	62.3	87.1	92.8	71.5	91.2	95.2	48.4	76.7	85.9
	UNITER-Base	~900	85.9	97.1	98.8	64.4	87.4	93.1	72.5	92.4	96.1	50.3	78.5	87.2
	OSCAR-Base†	~900	-	-	-	70.0	91.1	95.5	-	-	-	54.0	80.8	88.5
	VinVL-Base†‡	~650	-	-	-	74.6	92.6	96.3	-	-	-	58.1	83.2	90.1
Grid	Pixel-BERT-X152	~160	87.0	98.9	99.5	63.6	87.5	93.6	71.5	92.1	95.8	50.1	77.6	86.2
	Pixel-BERT-R50	~60	75.7	94.7	97.1	59.8	85.5	91.6	53.4	80.4	88.5	41.1	69.7	80.5
Linear	ViLT-B/32	~15	81.4	95.6	97.6	61.8	86.2	92.6	61.9	86.8	92.8	41.3	72.0	82.5
	ViLT-B/32ⓐ	~15	83.7	97.2	98.1	62.9	87.1	92.7	62.2	87.6	93.2	42.6	72.8	83.4
	ViLT-B/32ⓐ⊕	~15	83.5	96.7	98.6	61.5	86.3	92.7	64.4	88.7	93.8	42.7	72.9	83.1

Natural Language for Visual Reasoning. The NLVR2 task is a binary classification task given triplets of two images and a question in natural language. As there are two input images unlike the pre-training setup, multiple strategies exist[7]. Following OSCAR (Li et al., 2020b) and VinVL (Zhang et al., 2021), we use the *pair* method. Here, the triplet input is reformulated into two pairs (question, image1) and (question, image2), and each pair goes through the ViLT. The head takes the concatenation of two pooled representations (p) as input and outputs the binary prediction.

Table 2 shows the results. ViLT-B/32 maintains competitive performance on both datasets considering its remarkable inference speed.

4.4. Retrieval Tasks

We fine-tune ViLT-B/32 on the Karpathy & Fei-Fei (2015) split of MSCOCO and F30K. For image-to-text and text-to-image retrieval, we measure both zero-shot and fine-tuned performance[8]. We initialize the similarity score head from the pre-trained ITM head, particularly the part that computes the true-pair logits. We sample 15 random texts as negative samples and tune the model with cross-entropy loss that maximizes the scores on positive pairs.

We report the zero shot retrieval results in Table 3 and fine-tuned results in Table 4. At zero-shot retrieval, ViLT-B/32 performs better in general than ImageBERT despite ImageBERT's pre-training on a larger (14M) dataset. At fine-tuned retrieval, recalls for ViLT-B/32 are higher by a large margin than the second fastest model (Pixel-BERT-R50).

4.5. Ablation Study

In Table 5, we perform various ablations. More training steps, whole word masking, and image augmentation come to be beneficial, whereas an additional training objective does not help.

It has been reported that the number of training iterations affects the performance of self-supervised models (Devlin et al., 2019; Chen et al., 2020a;b). As VLP is also a form of self-supervised training, we examine the effects of training durations. As expected, the performance constantly increases as we train the model for longer training steps (rows 1~3). Masking whole words for the MLM objective (rows 3~4) and fine-tuning with augmentation (row 6) also

[7] UNITER proposed three downstream head setups: pair, triplet, and pair-biattn.

[8] R@K corresponds to whether the ground truth is included among top K results from the validation set.

Table 5. Ablation study of ViLT-B/32. ⓦ denotes whether whole word masking is used for pre-training. ⓜ denotes whether MPP objective is used for pre-training. ⓐ denotes whether RandAugment is used during fine-tuning.

Training Steps	Ablation ⓦ	ⓜ	ⓐ	VQAv2 test-dev	NLVR2 dev	NLVR2 test-P	Flickr30k R@1 (1K) TR (ZS)	Flickr30k R@1 (1K) IR (ZS)	MSCOCO R@1 (5K) TR (ZS)	MSCOCO R@1 (5K) IR (ZS)
25K	X	X	X	68.96 ± 0.07	70.83 ± 0.19	70.83 ± 0.23	75.39 (45.12)	52.52 (31.80)	53.72 (31.55)	34.88 (21.58)
50K	X	X	X	69.80 ± 0.01	71.93 ± 0.27	72.92 ± 0.82	78.13 (55.57)	57.36 (40.94)	57.00 (39.56)	37.47 (27.51)
100K	X	X	X	70.16 ± 0.01	73.54 ± 0.02	74.15 ± 0.27	79.39 (66.99)	60.50 (47.62)	60.15 (51.25)	40.45 (34.59)
100K	O	X	X	70.33 ± 0.01	74.41 ± 0.21	74.57 ± 0.09	81.35 (69.73)	61.86 (51.28)	61.79 (53.40)	41.25 (37.26)
100K	O	O	X	70.21 ± 0.05	72.76 ± 0.50	73.54 ± 0.47	78.91 (63.67)	58.76 (46.96)	59.53 (47.75)	40.08 (32.28)
100K	O	X	O	70.85 ± 0.13	74.91 ± 0.29	75.57 ± 0.61	83.69 (69.73)	62.22 (51.28)	62.88 (53.40)	42.62 (37.26)
200K	O	X	O	71.26 ± 0.06	75.70 ± 0.32	76.13 ± 0.39	83.50 (73.24)	64.36 (54.96)	61.49 (56.51)	42.70 (40.42)

Table 6. Comparison of VLP models in terms of parameter size, FLOPs, and inference latency. Since FLOPs are proportional to input size, we denote the number of input tokens (image+text) as superscripts ("?" when text length is unreported; we arbitrarily use length 40). Although not captured in FLOPs count nor parameter size (because it is not a tensor operation), note that *per-class* NMS for 1,600 classes amounts to more than 500 ms in latency. NMS latency varies a lot according to the number of detected classes.

Visual Embed	Model	#Params (M)	#FLOPs (G)	Time (ms)
Region	ViLBERT^{36+36}	274.3	958.1	~900
	VisualBERT^{36+128}	170.3	425.0	~925
	LXMERT^{36+20}	239.8	952.0	~900
	UNITER-Base^{36+60}	154.7	949.9	~900
	OSCAR-Base^{50+35}	154.7	956.4	~900
	VinVL-Base^{50+35}	157.3	1023.3	~650
	Unicoder-VL$^{100+?}$	170.3	419.7	~925
	ImageBERT^{100+44}	170.3	420.6	~925
Grid	Pixel-BERT-X152$^{146+?}$	144.3	185.8	~160
	Pixel-BERT-R50$^{260+?}$	94.9	136.8	~60
Linear	ViLT-B/32^{200+40}	87.4	55.9	~15

drive performance. Further increase in training iterations to 200K improved performance on VQAv2, NLVR2, and zero-shot retrieval. We stop increasing the number of iterations over 200K as the fine-tuned text retrieval performance decreases afterward.

An additional masked region modeling (MRM) objective has been the key for performance boost in VLP models such as Chen et al. (2019). We experiment with masked patch prediction (MPP) (Dosovitskiy et al., 2020) which mimics the effect of MRM in a form compatible with patch projections. The patch v is masked with the probability of 0.15, and the model predicts the mean RGB value of the masked patch from its contextualized vector $z^D_{\text{masked}|v}$. However, MPP turns out not to be contributing to downstream performance (rows 4~5). This result is in sharp contrast to the MRM objective on supervision signals from object detection.

Table 7. VLP model components. "PC" is for *per-class* manner NMS and "CA" is for *class-agnostic*. Following Tan & Bansal (2019), one single-modality layer is counted as 0.5 multi-modality layer.

Visual Embed	Model	CNN Backbone	RoI Head	NMS	Trans. Layers
Region	ViLBERT	R101	C4	PC	~15
	VisualBERT	X152	FPN	PC	12
	LXMERT	R101	C4	PC	~12
	UNITER-Base	R101	C4	PC	12
	OSCAR-Base	R101	C4	PC	12
	VinVL-Base	X152	C4	CA	12
	Unicoder-VL	X152	FPN	PC	12
	ImageBERT	X152	FPN	PC	12
Grid	Pixel-BERT-X152	X152	-	-	12
	Pixel-BERT-R50	R50	-	-	12
Linear	ViLT-B/32	-	-	-	12

4.6. Complexity Analysis of VLP Models

We analyze the complexity of VLP models in various terms. In Table 6, we report the number of parameters, the number of floating-point operations (FLOPs), and the inference latency of the visual embedder and transformer. We exclude the textual embedder because it is shared by all VLP models[9]. The latency is averaged over 10K times on a Xeon E5-2650 CPU and an NVIDIA P40 GPU.

The input size in terms of image resolution and the length of concatenated multimodal input sequence affects the number of FLOPs. We co-note the sequence lengths. The image resolution is 800 $\times$ 1,333 for region-based VLP models and Pixel-BERT-R50, 600 $\times$ 1,000 for Pixel-BERT-X152, and 384 $\times$ 640 for ViLT-B/32.

In Pixel-BERT and ViLT, visual tokens are sampled during pre-training and used in full during fine-tuning. We report the maximum number of visual tokens.

We observe that the runtime of BERT-base-like transformers varies only by $<$ 1 ms for input sequences of length under 300. Since patch projection of ViLT-B/32 generates at most

[9]FLOPs and time are neglectable because the operation is an embedding lookup. The 30K embedding dictionary used by `bert-base-uncased` has 23.47 M parameters

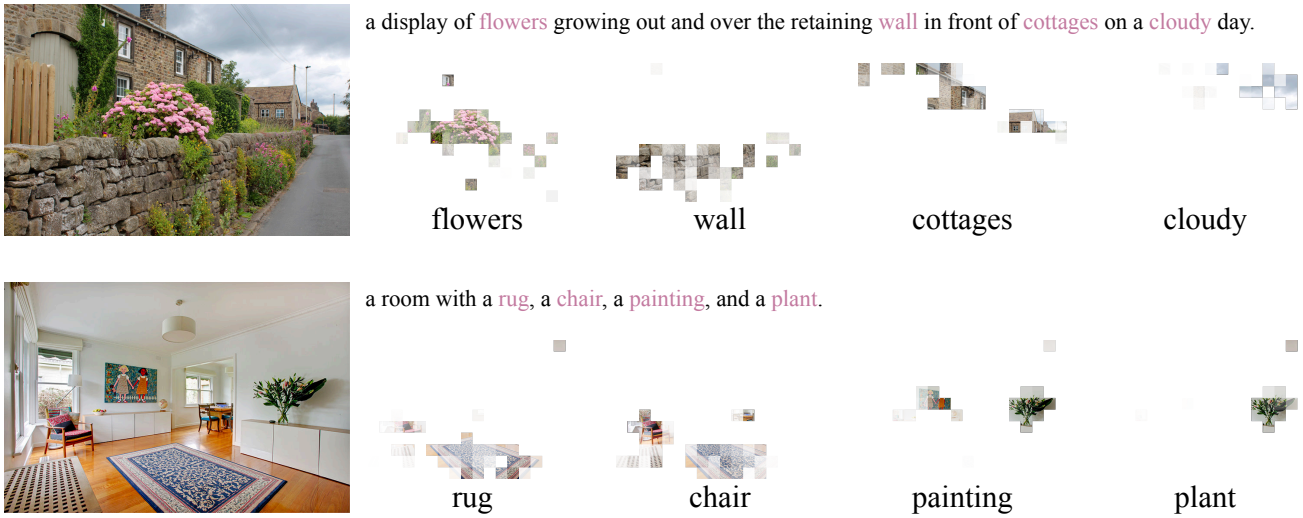

Figure 4. Visualizations of transportation plan of word patch alignment. Best viewed zoomed in.

240 image tokens, our model can still be efficient even though it receives a combination of image and text tokens.

4.7. Visualization

Figure 4 is an example of a cross-modal alignment. The transportation plan of WPA expresses a heatmap for a text token highlighted in pink color. Each square tile represents a patch, and its opacity indicates how much mass is transported from the highlighted word token.

More IPOT iterations– more than over 50 as in the training phase– help the visualization heatmap converge; empirically, 1,000 iterations are sufficient to get a clearly identifiable heatmap. We z-normalize the plan for each token and clamp the values to [1.0, 3.0].

5. Conclusion and Future Work

In this paper, we present a minimal VLP architecture, Vision-and-Langauge Transformer (ViLT). ViLT is competent to competitors which are heavily equipped with convolutional visual embedding networks (e.g., Faster R-CNN and ResNets). We ask for future work on VLP to focus more on the modality interactions inside the transformer module rather than engaging in an arms race that merely powers up unimodal embedders.

Although remarkable as it is, ViLT-B/32 is more of a proof of concept that efficient VLP models free of convolution and region supervision can still be competent. We wrap up by pointing out a few factors that may add to the ViLT family.

Scalability. As shown in papers on large-scale transformers (Devlin et al., 2019; Dosovitskiy et al., 2020), the performance of pre-trained transformers scale well given an appropriate amount of data. This observation paves the way for even better performing ViLT variants (e.g., ViLT-L (large) and ViLT-H (huge)). We leave training larger models for future work because aligned vision-and-language datasets are yet scarce.

Masked Modeling for Visual Inputs. Considering the success of MRM, we speculate that the masked modeling objective for the visual modality helps by preserving the information up until the last layer of the transformer. However, as observed in Table 5, a naive variant of MRM on image patches (MPP) fails.

Cho et al. (2020) proposed to train their grid RoIs on masked object classification (MOC) tasks. However, the visual vocabulary cluster in this work was fixed during the vision and language pre-training together with the visual backbone. For trainable visual embedders, one-time clustering is not a viable option. We believe that alternating clustering (Caron et al., 2018; 2019) or simultaneous clustering (Asano et al., 2019; Caron et al., 2020) methods studied in visual unsupervised learning research could be applied.

We encourage future work that does not use region supervision to devise a more sophisticated masking objective for the visual modality.

Augmentation Strategies. Previous work on contrastive visual representation learning (Chen et al., 2020a;b) showed that gaussian blur, not employed by RandAugment, brings noticeable gains to downstream performance compared with a simpler augmentation strategy (He et al., 2020). Exploration of appropriate augmentation strategies for textual and visual inputs would be a valuable addition.

References

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

Asano, Y., Rupprecht, C., and Vedaldi, A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2019.

Berman, M., Jégou, H., Vedaldi, A., Kokkinos, I., and Douze, M. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

Bugliarello, E., Cotterell, R., Okazaki, N., and Elliott, D. Multimodal pretraining unmasked: Unifying the vision and language berts. *arXiv preprint arXiv:2011.15124*, 2020.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Caron, M., Bojanowski, P., Mairal, J., and Joulin, A. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2959–2968, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019.

Cho, J., Lu, J., Schwenk, D., Hajishirzi, H., and Kembhavi, A. X-lxmert: Paint, caption and answer questions with multi-modal transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8785–8805, 2020.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., and Hu, G. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

Gan, Z., Chen, Y.-C., Li, L., Zhu, C., Cheng, Y., and Liu, J. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*, 2020.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., and Soudry, D. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8129–8138, 2020.

Huang, Z., Zeng, Z., Liu, B., Fu, D., and Fu, J. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E., and Chen, X. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10267–10276, 2020.

Jiang, Y., Natarajan, V., Chen, X., Rohrbach, M., Batra, D., and Parikh, D. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

Karpathy, A. and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. The open images dataset v4. *International Journal of Computer Vision*, pp. 1–26, 2020.

Lee, K.-H., Chen, X., Hua, G., Hu, H., and He, X. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–216, 2018.

Li, G., Duan, N., Fang, Y., Gong, M., Jiang, D., and Zhou, M. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pp. 11336–11344, 2020a.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pp. 121–137. Springer, 2020b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Lu, J., Batra, D., Parikh, D., and Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pp. 13–23, 2019.

Lu, J., Goswami, V., Rohrbach, M., Parikh, D., and Lee, S. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10437–10446, 2020.

Nguyen, D.-K., Goswami, V., and Chen, X. Revisiting modulated convolutions for visual counting and beyond. *arXiv preprint arXiv:2004.11883*, 2020.

Ordonez, V., Kulkarni, G., and Berg, T. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24:1143–1151, 2011.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Qi, D., Su, L., Song, J., Cui, E., Bharti, T., and Sacheti, A. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020.

Radford, A., Sutskever, I., Kim, J., Krueger, G., and Agarwal, S. Learning transferable visual models from natural language supervision, 2021.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018.

Tan, H. and Bansal, M. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*, pp. 433–453. PMLR, 2020.

Yu, F., Tang, J., Yin, W., Sun, Y., Tian, H., Wu, H., and Wang, H. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*, 2020.

Yu, Z., Yu, J., Cui, Y., Tao, D., and Tian, Q. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6281–6290, 2019.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. Vinvl: Making visual representations matter in vision-language models. *arXiv preprint arXiv:2101.00529*, 2021.

Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J. J., and Gao, J. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, pp. 13041–13049, 2020.