MiniVLM: A Smaller and Faster Vision-Language Model

Jianfeng Wang Xiaowei Hu Pengchuan Zhang Xiujun Li Lijuan Wang
Lei Zhang Jianfeng Gao Zicheng Liu
Microsoft
{jianfw,xiaowh,penzhan,xiul,lijuanw,leizhang,jfgao,zliu}@microsoft.com

Abstract

Recent vision-language (VL) studies have shown remarkable progress by learning generic representations from massive image-text pairs with transformer models and then fine-tuning on downstream VL tasks. While existing research has been focused on achieving high accuracy with large pre-trained models, building a lightweight model is of great value in practice but is less explored. In this paper, we propose a smaller and faster VL model, MiniVLM, which can be finetuned with good performance on various downstream tasks like its larger counterpart. MiniVLM consists of two modules, a vision feature extractor and a transformer-based vision-language fusion module. We design a Two-stage Efficient feature Extractor (TEE), inspired by the one-stage EfficientDet [48] network, to significantly reduce the time cost of visual feature extraction by 95%*, compared to a baseline model. We adopt the MiniLM [54] structure to reduce the computation cost of the transformer module after comparing different compact BERT models. In addition, we improve the MiniVLM pre-training by adding* $7M$ *Open Images data, which are pseudo-labeled by a state-of-the-art captioning model. We also pre-train with high-quality image tags obtained from a strong tagging model to enhance cross-modality alignment. The large models are used offline without adding any overhead in fine-tuning and inference. With the above design choices, our MiniVLM reduces the model size by* 73% *and the inference time cost by* 94% *while being able to retain* $94-97\%$ *of the accuracy on multiple VL tasks. We hope that MiniVLM helps ease the use of the state-of-the-art VL research for on-the-edge applications.*

1. Introduction

With the success of BERT [5] and recent advances [59, 32, 43, 46, 22, 21, 23, 10, 3, 12, 24] in vision-language pre-training (VLP), models pre-trained on large-scale image-text data have made substantial improvement on various benchmarks for a wide range of vision-language (VL) tasks, such as image captioning, visual question answering and image-text retrieval. The models used in most VLP works contain two modules: the vision module based on convolutional neural networks trained on ImageNet [39] and/or Visual Genome (VG) [17] to extract visual features from the image; and the feature fusion module based on the multi-modal transformer model to process both the visual features generated by the vision module and the token embeddings of the text input. The VL models are firstly pre-trained to learn cross-modal representations, and then fine-tuned on task-specific data. In recent VLP research, both of the two modules leverage large-scale deep neural networks, which can take up to hundreds of millions of parameters, to reach the state-of-the-art performance. However, due to the large sizes and high computation cost at inference time, it could be impractical for real-world applications to exploit the power of large models under a constrained training and/or inference budget. In fact, building a lightweight VL

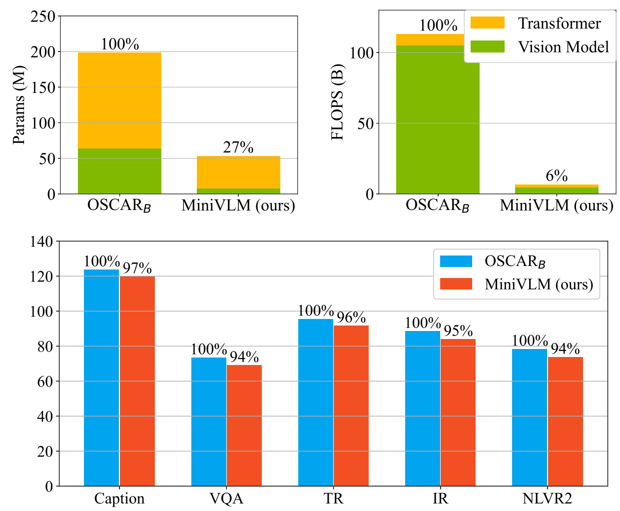

Figure 1: MiniVLM retains $94-97\%$ of the accuracy on multiple tasks with 27% parameters and 6% FLOPS compared to state-of-the-art model OSCAR$_B$ [23]. Details can be found in Sec. 4.4.

model, which is desired when operating on resource-limited devices, is of great value in practice but is less explored in the literature.

While larger models have been demonstrated to achieve higher performance in extensive studies, it is challenging to compress the model to smaller sizes without tremendous performance drop. In order to retain as much performance as possible, we firstly optimize the network architecture to balance the accuracy and speed. Moreover, we improve the small-model pre-training by leveraging large models and large-scale dataset.

For the architecture of VL models, popularized as "bottom-up top-down" (BUTD) attention [2], most existing works [59, 32, 43, 46, 22, 21, 23, 10, 3] use the ResNet-101 Faster R-CNN [38] model trained on the VG dataset as the visual feature extractor, which has been well validated by state-of-the-art results on various benchmarks. However, the detector suffers from a heavy model size and high latency, and consequently cannot be deployed to resource-limited applications. A few recent works [12, 24] revisit the usage of grid features from the convolutional layer to skip the region-related computation in Faster R-CNN. But it is still an open problem to select over the overwhelming number of grid features, as dumping the whole feature map to transformer could be prohibitively expensive in computation. For the transformer module, BERT is widely used as the de facto standard. Recent works in Natural Language Processing (NLP) have explored to maintain high performance with compact structures base on BERT. But there is still a lack of study on the compact structures in VLP.

In this paper, we propose a smaller and faster VL model, named MiniVLM, to reach similar performance as its larger counterpart with much smaller size, resulting in faster speed at inference time. For the vision module in MiniVLM, we design a **T**wo-stage **E**fficient feature **E**xtractors, named TEE, to drastically reduce the computation cost for extracting visual features, which is a dominating part of the inference cost on certain tasks. While refining each part of the detection model, we greatly simplify the region-related modules in TEE, specifically for the needs of VL tasks, which require rich visual representations rather than precise box locations as in the object detection task. Experimental results show that our TEE can extract similar quality visual features with smaller dimension sizes at a much faster speed. In particular, our TEE-0, using a similar backbone as EfficientDet-D0 [48], is $3.7\times$ smaller and $16.7\times$ faster than the widely used R101 Faster R-CNN from BUTD, while retaining competitive accuracy in detection on VG, and also up to 96% of the accuracy on downstream tasks. For the transformer model, we choose the MiniLM[54] structure after empirically evaluating the performance of several structures on multi-modal representation learning, including BERT [5] and its compact variants [40, 45, 15, 54].

In addition to the model architecture optimization, we leverage high-accuracy large models and large-scale data, either labeled or unlabeled, to further boost the performance of the small pre-trained model. To improve the accuracy of TEE, we pre-train it on large-scale classification and detection dataset before fine-tuning on VG. During VL pre-training, we apply data distillation [36] to add $7M$ Open Images [18] which are pseudo-labeled by the state-of-the-art "teacher" captioning model. We also use high precision tag prediction from a heavy tagging model in pre-training to improve visual-text alignment. The heavy tagging model is not used in fine-tuning or inference, and thus does not affect the runtime speed. With the above ingredients, our MiniVLM, composed of TEE-0 and MiniLM [54], reduces the end-to-end inference time to 6% with 27% parameters, and retains $94-97\%$ accuracy compared to large state-of-the-art models on multiple VL tasks.

In summary, we make the following contributions.

- We propose a VL model MiniVLM, which can be fine-tuned with good performance on multiple downstream tasks, while being smaller and faster for practical adoption.
- We design a Two-stage Efficient feature Extractor (TEE) to extract image region features for VL tasks, which generates features of good quality at a much faster speed.
- We demonstrate the benefits of using large models as well as large-scale data in the small VL model pre-training stage, which increases the CIDEr score by 4.8 in total in the downstream image captioning task.

2. Related work

Vision-Language Pre-training. Remarkable progress [32, 43, 12, 25, 22, 23, 10, 3] has been made recently on vision-language tasks through network pre-training on massive data with image-text pairs. A popular framework shared among most VLP works is to view the extracted visual features as visual 'tokens' and feed them together with text tokens into the BERT [5, 50] model for joint representation learning.

The visual feature is generally extracted with an off-the-shelf vision model, and the main focus is on the multi-modal fusion based on BERT model. With the multiple modalities, the fusion can be categorized as early fusion, late fusion and full fusion. Early fusion is to first process each modality separately and then to fuse them together, *e.g.* in ViLBERT [32], LXMERT [46], ERNIE-ViL [57]. Late fusion is to first process the two modalities together and then process each separately to enhance the single-modality task, *e.g.* InterBERT [25]. Full fusion means to process the two modalities' features together with the BERT

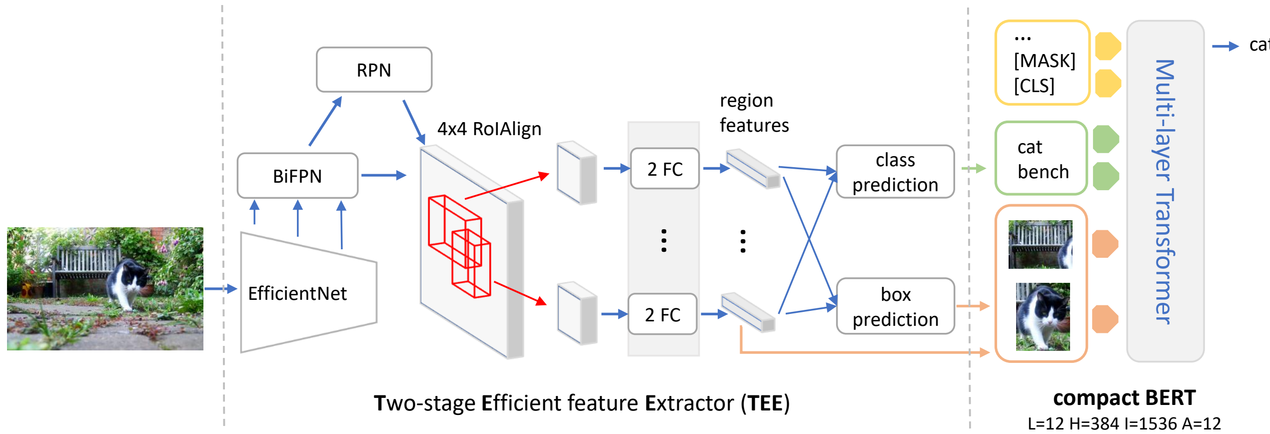

Figure 2: The proposed MiniVLM architecture, consisting of the Two-stage Efficient feature Extractor (TEE) and compact BERT feature fusion module. During inference, the vision module detects objects and extracts region features, which are fed to the transformer model. The text input of the transformer also depends on the downstream task. For image captioning, [CLS] is given as the first token, then the model predicts the next token in an auto-regressive manner to generate a caption sentence.

model from the very beginning to the final representation, *e.g*. OSCAR [23], Unicoder-VL [21], VL-BERT [43], UNITER [3], VIVO [10]. The pre-training tasks typically include the masked language modeling, image-text pairing loss, and masked region modeling.

Visual Feature Extractor. Visual feature extraction is one of the key modules in vision-language (VL) tasks. As in the bottom-up top-down approach [2], region features based on Faster-RCNN [38] have shown strong accuracy and been widely used in VL tasks [59, 32, 43, 46, 22, 21, 23, 10]. The extractor is built on ImageNet [39] and Visual Genome [17] datasets with two training tasks: one is to predict the object category and the other is to predict the attribute information.

An alternative approach is the grid feature, which is revisited in [14, 12] and demonstrated encouraging performance. In [14], the grid feature extractor is constructed by casting the Faster-RCNN model into a fully-convolutional network and remove the region-related operations (*e.g*. non-suppressed compression) to reduce the time cost. In [12], the convolutional network is trained together with modality fusion network without the detection data.

One advantage of using region features is that there is a simple way to select the top-K salient regions as each region is associated with a confidence score and nearby regions are also suppressed with the non-maximum suppression. Typically, the number of region features is 50 while the number of grid features ranges from 300 to 600 as in [14]. With more features, the cost of the multi-modal fusion part can be significantly increased. Thus, in this paper, we stick to the region features for our compact model.

Object Detection. Region feature is built on top of the object detector, and the detector can be two-stage [38, 8, 24, 55] or one-stage [37, 48, 29, 27, 6, 58, 53, 49]. The two-stage detector generates bounding box candidates with a region proposal network (RPN) and extracts the region features with RoIPool [38] or RoIAlign [8]. The feature is further processed with a classification head and a bounding box regression head. In contrast, the one-stage detector removes the RPN and feature extraction, and directly predicts the bounding box results based on the convolutional neural network.

Due to the removal of RPN and region feature extraction, fast object detector are mostly based on one-stage detectors, *e.g*. [37, 48, 52, 33, 20]. However, it remains open on how to effectively extract region features directly from one-stage detectors for VL models. Thus, we use a two-stage architecture but design a lightweight backbone and detection head for the compact VL model.

Compact BERT. $\text{BERT}_{\text{BASE}}$ or $\text{BERT}_{\text{LARGE}}$ has been commonly used in the existing VL works. To reduce the cost, one can simply reduce the network dimensions, *e.g*. the number of layers, the hidden size, as in TinyBERT [15] and MiniLM [54]. MobileBERT [45] constructs the network with the bottleneck design [9] to reduce the cost. ALBERT [19] focuses on the reduction of the parameter size. In our compact solution, after comparing different approaches in VL tasks, we choose MiniLM [54] as our multi-modal fusion module.

3. MiniVLM

In this section, we describe how we design a smaller and faster VL model, MiniVLM, and improve the small-model pre-training to retain high accuracy. An overview of our model is shown in Fig. 2. It consists of a detector-based feature extractor and a transformer-based feature fusion module. For different downstream tasks, we alter the transformer prediction head with minimal changes, which we defer to Sec. 4.4.

3.1. Model architecture

Two-stage Efficient feature Extractor. While the R101 Faster R-CNN detector from [2] has been widely used to extract region features, the computation cost is to a large extent overlooked, which can take up to 90% of the total inference time for some VL tasks. Even though region feature extraction is part of an objection detection model, the requirement for VL tasks is not the same as for objection detection. For VL tasks, the transformer helps in reasoning about both visual and language semantics, and what is needed from the feature extractor is rich visual representations. For example, the bounding box locations do not need to be highly accurate, and the recall of the bounding boxes is more important to cover more visual information from the image. These characteristics allow us to design a feature extractor that is much more efficient while at the same time does not cause significant accuracy degradation for the downstream tasks. Fig. 2 shows the design of our feature extractor called Two-stage Efficient feature Extractor (TEE).

First, we replace the backbone with EfficientNet [47] and add BiFPN [48] to generate multi-scale features. Both components are constructed with depthwise and pointwise convolutional layers, which reduce the model size and computation significantly compared with the standard convolutional layers. The BiFPN receives as input 4 layers with stride $= 4, 8, 16, 32$ from EfficientNet, and outputs 5 features with an extra feature map of stride $= 64$ by downsampling. Both EfficientNet and BiFPN are building blocks taken from the one-stage detector EfficientDet [48], while we make the change to use feature maps starting from stride $= 4$ instead of 8 to incorporate information from higher resolution feature maps for the feature extraction.

While region proposal network (RPN) [38] is used following the design of two-stage detectors, the box prediction modules are greatly simplified. Our RPN contains only 2 linear layers: one for bounding box regression and the other for objectness prediction. After non-maximal suppression (NMS) we select the feature map for each box proposal with heuristics from [26], and apply RoIAlign [8] operation, followed by 2 linear layers to extract the region features. The resolution of RoIAlign is reduced to 4×4 rather than 14×14 in [38, 2] or 7×7 in [26]. The feature's dimension is also reduced from 2048 [38, 2] to 1024. With the region features, two linear layers are used to predict the bounding box coordinates and class, respectively. Unlike [2], we apply NMS in a class-agnostic manner, which dramatically saves the computation. Otherwise, NMS is performed for each class in prediction, which could be as many as 1600 times on Visual Genome [17].

Similar to EfficientDet, we scale up the input image size, network depth and width to get stronger feature extractors. In our MiniVLM, we choose TEE-0, corresponding to EfficientDet-D0, for fast inference speed.

During inference, given an image $\mathbf{I}$, the vision module outputs a bag of region features $\mathbf{R}$ with corresponding bounding boxes $\mathbf{B}$ and class names $\mathbf{C}$, which are fed to the transformer model along with text tokens.

Multi-modal Transformer. With the extracted features, a transformer-based feature fusion module is applied. To make a good balance between speed and accuracy, we search the compact structures based on BERT by varying some parameters, *e.g.*, the number of layers. Based on experimental results, we choose the same structure as MiniLM [54], *i.e.*, 12 layers with hidden size reduced to 386 and feed forward intermediate size reduced to 1536. Following [23], the input to the transformer model consists of visual features formed by the concatenation of $\mathbf{R}$ and bounding box encoding (including normalized top-left, top-right, bottom-left, bottom-right and the box's width, height), tokenized object tag names $\mathbf{C}$, and tokenized sentences $\mathbf{S}$. The content of $\mathbf{S}$ can be different depending on the downstream task, *e.g.*, the question sentence for VQA, a single `[CLS]` token to indicate the start of sentence for image captioning.

3.2. Pre-training

To train a VL model, at first, the vision module is trained on classification or detection dataset to learn diverse visual representations. Then, given the visual features, the transformer module is pre-trained on massive image-text pairs to learn cross-modal representations. Finally, the model is fine-tuned on specific downstream tasks. To compensate the performance drop brought about by the small model size, we apply several techniques in training.

As visual features are critical in VL tasks, we enhance visual feature learning by pre-training TEE on large-scale classification and object detection dataset, *e.g.*, Objects365 [41], and then fine-tuning on the Visual Genome dataset, which demonstrates performance gain in downstream VL tasks.

By pre-training the transformer model on large-scale image-text data, our model inherits the advantage of VL pre-training. Moreover, we leverage large models in two ways to further exploit the potential for pre-training with compact VL models. First, we apply a state-of-the-art captioning model to automatically label $7M$ images from Open Images with pseudo captions. Thus, the small model learns to mimic the behavior of the large model through much more data, which can be further expanded with access to internet-scale unlabeled data. Second, we use a heavy

Model		Params(M)	FLOPS(B)	$mAP_{0.5}$
Grid [14]	R50	23.5	37.8	-
	X101	86.9	161.2	-
Region	R101-F [2]	63.8	104.8	10.2[1]
	TEE (ours)	7.5	4.4	9.9

Table 1: Comparison of different vision modules on number of parameters, FLOPS, and detection accuracy on VG.

Model	Config	Params(M)	FLOPS(B)
$\text{BERT}_{\text{BASE}}$ [5]	$12/768/3072$	134.3	8.2
BERT_8	$8/768/3072$	106.0	5.8
TinyBERT_6 [15]	$6/768/3072$	91.8	4.6
BERT_4	$6/768/3072$	77.6	3.3
MiniLM [54]	$12/384/1536$	45.7	2.3
TinyBERT_4 [15]	$4/312/1200$	24.3	0.8

Table 2: Computational cost for different transformer structures. Config: the number of layers, the embedding dimension, and intermediate size. FLOPs: measured in one forward pass with 50 image regions and 35 text tokens.

tagging model to generate high-quality tags, and also add ground truth tags if available. That is, during pre-training, the input tags can be from other sources. But in fine-tuning, the tags are generated by the same vision model used to extract features to remove the dependency on the heavy model at inference time. The experimental results in Sec. 4.5 shows the better quality of tags helps with cross-modal representation learning.

Other than the changes about the sources of object tags and the associated sentences, we use the same pre-training tasks as described in [23], including masked language modeling (MLM) and image-text (contrastive) matching (ITM). For fine-tuning and inference, we do not use large models. Hence, the improvement on pre-training benefits downstream tasks without adding any runtime overhead.

4. Experiment

4.1. Implementation details

TEE. We first pre-train the backbone with ImageNet [39] classification dataset, then pre-train the whole detection model with Objects365 [41], and lastly fine-tune it on Visual Genome [17]. On ImageNet, the backbone is trained from scratch for 400 epochs. Stochastic gradient descent (SGD) is used to optimize the model with the batch size of 1024. The learning rate is 0.4 and decays with a cosine scheduler [30]. Afterwards, the detection model is initialized with this ImageNet-pretrained backbone and trained on Objects365 for 100 epochs. The learning rate is 0.4 and batch size is 256 with SGD. Lastly, we fine-tune the model on Visual Genome for 200 epochs, with learning rate 0.2 and batch size 512. Following [2], an additional head is added to train with attribute classes.

Vision-Language Pre-training. We combine existing V+L datasets, including MS COCO [28], Conceptual Captions (CC) [42], SBU captions [34], Flicker30k [56], GQA [13], VQA [7] and VG-QA [17], resulting in 4 million unique images and 7 million text-image pairs as the pre-training corpus.

Moreover, to explore data distillation for compact VL model pre-training, we build a large-scale auto-captioned dataset based on $7M$ images from the Open Images V6[18]. We generate pseudo captions for each image using the state-of-the-art image captioning model fine-tuned from OSCAR [23]. The human verified positive tag ground truth labels are combined with object class prediction results from TEE-3, and together serve as the tag input in our VL pre-training. This dataset is referred to as OI-Caps-7M. We observe significant improvement when adding this dataset in pre-training. We will release this dataset and our pre-trained MiniVLM model to the public.

During vision-language pre-training, the batch size is 2048. The initial learning rate is 4×10^{-4} with linear decay. The model is updated with AdamW [31] optimizer for 100 epochs on the combined 7 million corpus and OI-Caps-7M.

4.2. Smaller and faster

As shown in Fig. 1, our MiniVLM reduces the number of parameters to 27% and FLOPS to 6% in total compared to the model in [23]. The following details the compression for both the vision module and the transformer module.

TEE. We compare our region feature extractor TEE with the widely used ResNet-101 Faster RCNN model (R101-F) from [2], as well as the grid feature extractor based on ResNet-50 and ResNeXt-101 from [14]. Table 1 shows the size and computation cost for each model at inference time. Compared to R101-F, our TEE reduces the number of parameters to $7.5/63.8 = 11.8\%$, and FLOPS to $3.3/89.4 = 3.7\%$. Our model also uses less parameters and FLOPS than grid feature extractors. The reason is the use of the depthwise and pointwise convolutional layers in backbone and the lightweight region feature extraction head.

On a CPU workstation[2] with 4 threads, with models implemented in PyTorch and processing one image at a time, Grid R50 takes 699.8 ± 110.1 ms, R101-F takes 12.3 ± 3.2 seconds, while our TEE takes only 393.9 ± 43.8 ms, which is 3.2% of R101-F. Note that inference speed highly depends on hardware and implementation, so we mainly report FLOPS for fair comparison.

[1]This number is from https://github.com/peteanderson80/bottom-up-attention.

[2]Intel(R) Xeon(R) CPU E5-2620 v4 @2.10GHz

Method	Network		Cost		Evaluation on Captioning			
	Vision	Transformer	Params(M)↓	FLOPs(B)↓	B@4↑	M↑	C↑	S↑
OSCAR_B [23]	R101-F	BERT_BASE	198.1	113.0	36.5	30.3	123.7	23.1
BERT → MiniLM	R101-F	MiniLM	109.5	107.1	35.0	28.5	118.8	21.7
R101-F → TEE-0	TEE-0	BERT_BASE	141.8	12.6	34.6	28.4	118.7	21.6
baseline	TEE-0	MiniLM	53.2	6.7	34.0	27.8	115.0	21.2
+O	TEE-0	MiniLM	53.2	6.7	34.3	28.1	116.7	21.3
+O + H	TEE-0	MiniLM	53.2	6.7	34.7	28.3	117.7	21.4
+O + H + DD (MiniVLM)	TEE-0	MiniLM	53.2	6.7	35.6	28.6	119.8	21.6

Table 3: Retaining high accuracy from OSCAR_B to MiniVLM on the image captioning task. Models are fine-tuned on COCO with cross entropy. O: using Objects365 to pre-train TEE before fine-tuning on Visual Genome. H: using high-quality tags during vision-language pre-training. DD: with data distillation by adding OI-Caps-7M to the pre-training corpus. For the metrics, ↑ indicates higher is better, ↓ indicates lower is better. Captioning results are evaluated with BLEU-4 (B@4), METEOR (M), CIDEr (C) and SPICE (S).

Compact BERT. Table 2 shows the comparison of different transformer model structures, including TinyBERT [15] with 4 and 6 layers, MiniLM [54], BERT_BASE with the number of layers cut to 4 (BERT_4) and 8 (BERT_8), and BERT_BASE. The whole model, including embedding and decoding layers, is counted. By choosing MiniLM, which will be discussed later with Fig. 4, the number of parameters is reduced to $45.7/134.3 = 34.0\%$, and FLOPS to $2.3/8.2 = 28.0\%$. As to the inference time evaluated on the image captioning task, BERT_BASE takes 712.6 ± 176.1 ms to process one image, while MiniLM takes 346.7 ± 49.8 ms, reducing to $346.7/712.6 = 48.7\%$.

4.3. Retaining high accuracy

Table 3 shows the ablation study on improving the pre-training for our small model. The pre-trained models are fine-tuned and evaluated on the image captioning task, which will be detailed in Sec. 4.4. The cost is measured end-to-end including both vision and transformer modules. Starting from the OSCAR_B [23] model, which consists of R101-F and BERT_BASE, if we replace the transformer module with MiniLM, the CIDEr score drops by 4.9. And if solely replacing the vision module with TEE-0, the CIDEr score drops similarly by 5.0. This also indicates that our feature extractor TEE-0 can achieve $118.7/123.7 = 96\%$ of the accuracy compared to R101-F on this task without any additional techniques. Then, we replace both modules to TEE-0 and MiniLM, respectively, which decrease the CIDEr score by 8.7. This is the baseline performance of our compact VL model.

On the baseline model, we apply approaches described in Sec. 3.2 and show the improvement for small-model pre-training. First, we use the Objects365 dataset to pre-train TEE before fine-tuning it on VG, which improves the CIDEr score by 1.7, indicating that better visual features contribute to better performance on VL tasks. Secondly, we use high-quality tags, generated from the stronger vision model TEE-3, during vision-language pre-training, and further improve the score by 1.0. The intuition is based on [23] that the tag information in pre-training helps with visual-text alignment. Lastly, we add the OI-Caps-7M dataset, which is automatically labeled by the large captioning model, and observe the gain of 2.1 in CIDEr. In total, the CIDEr score is improved by 4.8 with large models and data distillation, resulting in a much smaller gap with the large pre-trained VL model.

With above ingredients, compared to OSCAR_B, our MiniVLM uses 73% less parameters and 94% less FLOPS, while retaining 97% of the CIDEr score in image captioning task.

4.4. Results on downstream VL tasks

Image Captioning. The task is to describe an image with a natural language sentence. Following [59, 23], we fine-tune the model with region features, captioning tokens and object tags. Captioning tokens are randomly replaced by the token of `[MASK]` with 15% chance and predicted by the corresponding representation, the attention of which is only on region features, tags and preceding caption tokens. The training task is either the cross entropy loss or the loss optimized for the CIDEr [51] score, and we report results with both tasks. During inference, the `[MASK]` is appended recursively with the generated tokens to predict the next token one by one. Considering the inference speed, we reduce the beam search size to 1 instead of 5 as in [23]. The accuracy is evaluated with BLUE@4 [35], METEOR [4], CIDEr [51], and SPICE [1]. The experiments are conducted on COCO dataset.

VQA. The task [7] is to answer a question with natural language based on the image context. Following [23], we cast

Method	CE Optimization				CIDEr optimization			
	B@4	M	C	S	B@4	M	C	S
BUTD [2]	36.2	27.0	113.5	20.3	36.3	27.7	120.1	21.4
Grid [14]	36.4	27.4	113.8	20.7	-	-	-	-
AoANet [11]	37.2	28.4	119.8	21.3	38.9	29.2	129.8	22.4
OSCAR_B [23]	36.5	30.3	123.7	23.1	40.5	29.7	137.6	22.8
MiniVLM (Ours)	35.6	28.6	119.8	21.6	39.2	29.7	131.7	23.5

Table 4: Image captioning evaluation results (single model) on COCO 'Karpathy' [16] test split. (B@4: BLUE@4, M: METEOR, C: CIDEr, S: SPICE.)

Method	VQA		NLVR2	
	test-std	test-dev	Test-P	Dev
BUTD [2]	70.34	-	-	-
Grid [14]	-	72.59	-	-
Pixel. (R50) [12]	71.42	71.35	71.7	72.4
Pixel. (X152) [12]	74.55	74.45	76.5	77.2
VisualBERT [22]	71.00	70.80	67.4	67.0
OSCAR_B [23]	73.44	73.16	78.07	78.36
MiniVLM (Ours)	69.39	69.06	73.93	73.71

Table 5: VQA and NLVR2 evaluation results.

Method	1K test set						5K test set					
	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
PixelBERT (R50) [12]	77.8	95.4	98.2	64.1	91.0	96.2	53.4	80.4	88.5	41.1	69.7	80.5
PixelBERT (X152) [12]	84.9	97.7	99.3	71.6	93.7	97.4	63.6	87.5	93.6	50.1	77.6	86.2
$\text{Unicoder-VL}_\text{B}$ [21]	84.3	97.3	99.3	69.7	93.5	97.2	62.3	87.1	92.8	46.7	76.0	85.3
OSCAR_B [23]	88.4	99.1	99.8	75.7	95.2	98.3	70.0	91.1	95.5	54.0	80.8	88.5
MiniVLM (Ours)	81.1	96.1	99.2	68.5	93.0	97.1	58.8	85.1	91.7	45.0	74.1	84.0

Table 6: Image-Text Retrieval task evaluation results on COCO datasets.

the problem as a classification problem where each class corresponds to one answer. The model inference is the image region feature, the question token, and object tag tokens. The training is trained by the cross entropy loss over a shared set of 3129 answers. The inference is to select the answer with the highest confidence.

Natural Language Visual Reasoning for Real (NLVR2). The task's input is a pair of images and a natural description, and the goal [44] is to predict whether the description is true about the image pair. To fine-tune the network, we construct two input sequences, each containing the concatenation of the description and one image, and then two outputs corresponding to `[CLS]` are concatenated as the joint representation for a binary linear classifier.

Image-Text Retrieval. The task is to retrieve similar images based on the text description or vice versa. The key is to score the similarity of image-text pairs. The model is trained as a binary classification task where the input is the image region features and the associated or mismatched text description. The transformer output corresponding to `[CLS]` is used for binary judgement. The experiments are on COCO dataset, and we report top-K retrieval accuracy for both $1K$ test sets and $5K$ test sets.

Results. Table 4, 5, and 6 show the results on image captioning, VQA, NLVR2 and image-text retrieval, respectively. As summarized in Fig. 1, we retain $94-97\%$ of the accuracy on downstream tasks compared with the state-of-the-art model OSCAR_B. In the figure, captioning is measured by CIDEr score with cross entropy optimization on COCO. Text retrieval (TR) and image retrieval (IR) are on $5K$ test set, and measured by R@10. VQA is on the `test-std` split, and NLVR2 is on `test-P` split. For image captioning, the CIDEr score of OSCAR_B [23] is 123.7, while our MiniVLM achieves 119.8 CIDEr, reaching $119.8/123.7 = 97\%$ accuracy. Compared with [14], which uses X101 to extract the grid feature, our solution achieves even higher CIDEr (119.8 vs 113.8) with much lower (3.3 vs 161.2 in FLOPS) feature extraction cost as shown in Table. 1. On NLVR2 and image-text retrieval, our MiniVLM achieves higher scores than [12] which uses ResNet50 as the grid feature extractor, while both our vision and transformer modules are smaller.

4.5. Analysis

In this section, we provide analysis on the model components and vision-language pre-training methods. Results are based on the models pretrained on the $7M$ corpus without OI-Caps-7M. All the models are fine-tuned on the COCO image captioning task, and the CIDEr score is reported.

Impact of Object Tags in Pre-training. While Table 3 has shown that using tags of higher quality can help to improve the small-model pre-training, we study the impact of tags under more settings as shown in Table 7. For the vi-

Feature	Init	Tagging Model		
		No Tag	TEE-S	TEE-3
TEE-0	Text	113.5	117.2	117.2
	Random	114.7	116.7	117.7
TEE-1	Text	118.5	118.9	119.5
	Random	118.6	119.1	120.5

Table 7: Impact of the tag input used in pre-training, comparing no tag with tags predicted by the same vision module (TEE-S), and tags predicted with higher quality by a stronger model (TEE-3). During fine-tuning, tags (except No Tag) are predicted by the same vision module to keep consistent as inference time. "Text" means the model is initialized from text pre-trained weights provided by [54]. "Random" means the model is initialized from scratch.

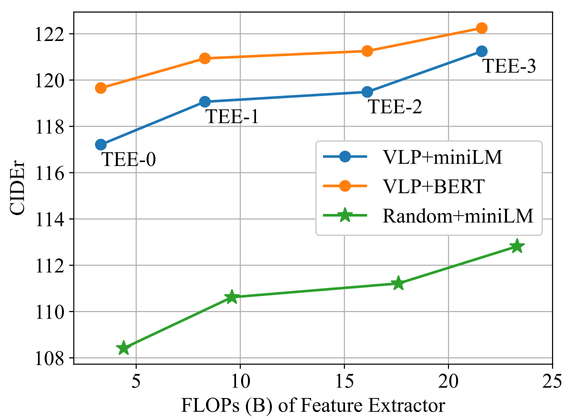

Figure 3: Impact of the backbone in TEE. Overall, the stronger feature extractor leads to the higher score.

sion module, we use TEE-0 or its stronger version TEE-1. The transformer module, which adopts the structure of MiniLM, is either initialized randomly or from the text pre-trained weights of MiniLM, which is developed for NLP tasks. From the results, using tags as input in pre-training makes larger improvement for the smaller model (TEE-0), while it consistently improves under all settings, and using better tags leads to better results. For small models, it might be harder to learn good representations, and thus the tag can contribute more in cross-modal alignment. Another observation is that random initialization gives comparable or even better results than the text-pretrained weights initialization. This is similar to the findings in [46].

Varying the backbone of TEE. To study the impact of vision modules, we scale up TEE, ranging from TEE-0 to TEE-3 with larger sizes and better detection accuracy as shown in Table 8. For each vision module, we combine with MiniLM or BERT. As shown in Fig. 3, stronger vision module leads to better accuracy, for small or large transformer

Model	Params (M)	FLOPS (B)	$mAP_{0.5}$
TEE-0	7.5	4.4	9.9
TEE-1	10.6	9.6	10.6
TEE-2	12.4	17.6	11.3
TEE-3	17.0	23.3	11.5

Table 8: Performance of different variants of our detectors. A larger backbone gives higher accuracy, but more costs.

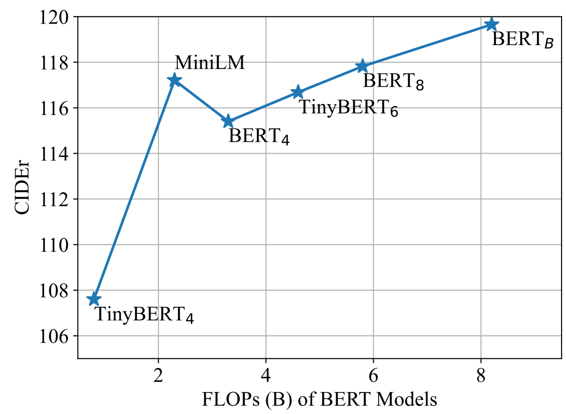

Figure 4: Experimental results of different pre-trained models in search of a good compact BERT structure.

models.

Impact of Compact BERT Structures. Fig. 4 shows the experimental results on speed-accuracy trade-off for models with different transformer modules listed in Table 2. Among the structures, MiniLM achieves a better trade-off between speed and accuracy. Note that the structure of TinyBERT$_6$ is the same as cutting the number of layers in BERT to 6. This shows that a "thinner" version of BERT could make better trade-off than the "shallower" version for VL tasks.

5. Conclusion

In this paper, we have proposed a compact solution, MiniVLM, for vision-language (VL) tasks, which is smaller and faster, and thus can be deployed in real-world applications on resource-constrained devices. For the vision module, we design the Two-stage Efficient feature Extractor (TEE), to significantly save computation by simplifying the region head and replacing regular convolutional layers with pointwise and depthwise convolution layers. To improve the small-model pre-training, we leverage large models and large-scale dataset. We fine-tune the pre-trained model on various downstream VL tasks, and show that MiniVLM can retain $94 - 97\%$ of the accuracy with 27% parameters and 6% FLOPS compared to the state-of-the-art VL model.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 382–398. Springer, 2016. 6

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. IEEE Computer Society, 2018. 2, 3, 4, 5, 7

[3] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: universal image-text representation learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, volume 12375 of *Lecture Notes in Computer Science*, pages 104–120. Springer, 2020. 1, 2, 3

[4] Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 376–380. The Association for Computer Linguistics, 2014. 6

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. 1, 2, 5

[6] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. NAS-FPN: learning scalable feature pyramid architecture for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7036–7045. Computer Vision Foundation / IEEE, 2019. 3

[7] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837, 2016. 5, 6

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):386–397, 2020. 3, 4

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 3

[10] Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. VIVO: surpassing human performance in novel object captioning with visual vocabulary pre-training. *CoRR*, abs/2009.13682, 2020. 1, 2, 3

[11] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *International Conference on Computer Vision*, 2019. 7

[12] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *CoRR*, abs/2004.00849, 2020. 1, 2, 3, 7

[13] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6700–6709. Computer Vision Foundation / IEEE, 2019. 5

[14] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020. 3, 5, 7

[15] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351, 2019. 2, 3, 5, 6

[16] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137. IEEE Computer Society, 2015. 7

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332, 2016. 1, 3, 4, 5

[18] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 2, 5

[19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019. 3

[20] Hei Law, Yun Teng, Olga Russakovsky, and Jia Deng. Cornernet-lite: Efficient keypoint based object detection. *CoRR*, abs/1904.08900, 2019. 3

[21] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium*

on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 11336–11344. AAAI Press, 2020. 1, 2, 3, 7

[22] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019. 1, 2, 3, 7

[23] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, volume 12375 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2020. 1, 2, 3, 4, 5, 6, 7

[24] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head R-CNN: in defense of two-stage object detector. *CoRR*, abs/1711.07264, 2017. 1, 2, 3

[25] Junyang Lin, An Yang, Yichang Zhang, Jie Liu, Jingren Zhou, and Hongxia Yang. Interbert: Vision-and-language interaction for multi-modal pretraining. *arXiv preprint arXiv:2003.13198*, 2020. 2

[26] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944. IEEE Computer Society, 2017. 4

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):318–327, 2020. 3

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 5

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, volume 9905 of *Lecture Notes in Computer Science*, pages 21–37. Springer, 2016. 3

[30] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 5

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 5

[32] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13–23, 2019. 1, 2, 3

[33] Sachin Mehta, Mohammad Rastegari, Linda G. Shapiro, and Hannaneh Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9190–9200. Computer Vision Foundation / IEEE, 2019. 3

[34] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1143–1151, 2011. 5

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002. 6

[36] Ilija Radosavovic, Piotr Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4119–4128. IEEE Computer Society, 2018. 2

[37] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016. 3

[38] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. 2, 3, 4

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 1, 3, 5

[40] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. 2

[41] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *2019 IEEE/CVF International Conference on Computer Vision,*

ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pages 8429–8438. IEEE, 2019. 4, 5

[42] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565. Association for Computational Linguistics, 2018. 5

[43] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: pre-training of generic visual-linguistic representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 1, 2, 3

[44] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6418–6428. Association for Computational Linguistics, 2019. 7

[45] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2158–2170. Association for Computational Linguistics, 2020. 2, 3

[46] Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5099–5110. Association for Computational Linguistics, 2019. 1, 2, 3, 8

[47] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019. 4

[48] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10778–10787. IEEE, 2020. 1, 2, 3, 4

[49] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: A simple and strong anchor-free object detector. *CoRR*, abs/2006.09214, 2020. 3

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 2

[51] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society, 2015. 6

[52] Robert J. Wang, Xiang Li, Shuang Ao, and Charles X. Ling. Pelee: A real-time object detection system on mobile devices. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. 3

[53] Tiancai Wang, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Learning rich features at high-speed for single-shot object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1971–1980. IEEE, 2019. 3

[54] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *CoRR*, abs/2002.10957, 2020. 1, 2, 3, 4, 5, 6, 8

[55] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10183–10192. IEEE, 2020. 3

[56] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguistics*, 2:67–78, 2014. 5

[57] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *CoRR*, abs/2006.16934, 2020. 2

[58] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9756–9765. IEEE, 2020. 3

[59] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13041–13049. AAAI Press, 2020. 1, 2, 3, 6