Image Retrieval with Short Text Queries

Ojas Rane[1,*], Conor Nugent[2] and Brian Mac Namee[1]

[1]School of Computer Science, University College Dublin, Dublin, Ireland
[2] Shutterstock Ireland

Abstract

The development of multimodal embedding spaces has made the development of semantic image retrieval applications possible. Such embeddings are trained on sentence-length descriptions of images which provides the model with rich context and helps in forming strong joint representations. However, this creates a mismatch with real-world usage, as users generally query these systems with just short phrases or keywords. The performance of models trained using long text can be poor for short text queries. To address this issue, we propose an image retrieval system that works with short text queries, but still takes advantage of large models pre-trained on long captions. We use four methods to improve performance: Fine-Tuning, Prompting, Expanding Short Texts Using Pre-trained GPT, and the SILC framework. We conduct our experiments using two pre-trained models, CLIP and SigLIP, to evaluate the effectiveness of these methods in enhancing short text-based image retrieval; and using the Flickr30k dataset for which short search strings have been generated using a pre-trained BLIP model.

Keywords

image retrieval, machine learning, multi-modal models

1. Introduction

Embedding based Image retrieval is a fundamental computer-vision task that handles two modalities: language (the search terms entered by a user) and vision (the relevant images retrieved). The basic idea is to find similar images based on query text from a large database. Embedding based Image retrieval systems, like others in multi-modal machine learning tasks, have greatly benefited from the emergence of pre-trained models. Pre-trained models are trained on generic tasks using large datasets and the representations they learn can be used and fine-tuned for downstream tasks.

Initially, pre-training was primarily used in for text and images independently [1]. This started to change with the emergence of multi-modal pre-trained models like ViLBERT [2] , LXMERT [3], and UNITER [4]. These models take an image and associated text as input and pass them through independent image and text encoders. The resulting representations are then fused and passed through another transformer which generates a joint representation. The CLIP [5] model illustrated the usefulness of pre-training having been pre-trained on a dataset of 400 million image-text pairs.

Many of the pre-trained multi-modal models use datasets such as COCO [6], WIT [7], and Flickr [8], which primarily contain sentence-length text data. The longer text descriptions offer richer context, providing information not just about the objects in the corresponding image, but also about the background and scene details. This additional context helps form stronger joint representations, as the model can more easily match the correct image-text pairs. -> In real-world image retrieval systems, however, users typically input short, relatively unstructured text queries—often consisting of just a few keywords. Since these models are pre-trained on sentence-length data, they often fail to retrieve relevant output. The primary challenge with short text queries is their lack of context. When given only a brief description of an image, the model finds it difficult to accurately match the query with the correct image

To develop an embedding based image retrieval system that works effectively with short text queries, we have utilized pre-trained CLIP and SigLIP [9] models. These models are chosen for their strong zero-shot and generalization capabilities, which are particularly useful when working with limited

AICS'24: 32nd Irish Conference on Artificial Intelligence and Cognitive Science, December 09–10, 2024, Dublin, Ireland
*Corresponding author.
ojas.rane@ucdconnect.ie (O. Rane); cnugent@shutterstock.com (C. Nugent); brian.macnamee@ucd.ie (B. Mac Namee)

textual input. We evaluate four modifications to enhance effectiveness of image retrieval with short text queries: Fine-Tuning, Prompting, Expanding Short Texts Using Pre-Trained GPT Model [10], and SILC [11]. These techniques modify the text query to add more context before it is processed by the text encoder, thereby improving the system's ability to match short queries with relevant images. As for short text dataset, there is no available dataset with an image-text pair that contains short text. Hence we have implemented image captioning using a pre-trained BLIP [12] model on the Flickr30k dataset to generate short text. The experiments described in this paper provide a comprehensive analysis of how image retrieval systems built using CLIP and SigLIP [9] perform with short and long texts and how the four modifications help bridge the performance gap between short and long text queries in these models.

2. Related Work

The concept of pre-training has long been used in the field of computer vision, for example pre-trained models such as VGG16 [13] and ResNet [14]. These models showed the effectiveness of transfer learning for visual tasks. It wasn't long before researchers used similar techniques for language [1], followed by the emergence of the transformer architecture for language models [15], and then more powerful pre-trained language models such as BERT [16] and GPT [17]. The success of pre-trained models for both vision and language led to the development of multimodal models that combine image and text data.

Multi-modal Vision Language Pre-training (VLP) [18] has been used for tasks including visual question answering, image-text retrieval, and image captioning. These models are first pre-trained on a large dataset of aligned image-text pairs to form a joint representation of image-text dat, and then fine-tuned for specific downstream tasks. VLP models can be divided based on how they combine their image and text features, either early or late fusion. Early fusion involves combining features at the beginning of the pipeline, allowing the model to learn a joint representation from the start. In late fusion, modalities are processed separately before being combined at the decision level.

Early fusion models typically follow the BERT architecture, and can be further sub-divided into single-stream and dual-stream architectures. In a single-stream architecture like Pixel-BERT [19], VisualBERT [20], and UNITER [4], both visual and language modalities are processed together through a single transformer. In contrast, a dual-stream architecture like ViLBERT or LXMERT processes visual and language modalities separately using distinct transformers before combining them.

In late fusion models, modalities are processed separately before combination at the decision level. CLIP and ALIGN [21] are examples of late fusion models in vision-language pre-training. These architectures process image-text pairs independently before employing contrastive learning techniques to learn how they can be combined. SigLIP is a model similar to CLIP that has shown better performance with more computational efficiency.

Embedding based image retrieval systems can broadly be divided into two categories: non-pretrained models and pre-trained models. Non-pretrained models include SCAN [22] and VSRN [23]. SCAN is an attention-based model which uses Faster R-CNN [24] to align objects and words together. Pre-trained models have evolved over time. Earlier versions were built using architectures such as UNITER, Pixel-BERT, and ViLBERT. Currently, state-of-the-art performance in image retrieval is achieved using more advanced pre-trained models like ALIGN, FILIP [25], Florence [26], and BLIP-2 [27]

3. Modifications for Short Text Image Retreival

Pre-trained vision-language models like CLIP and SigLIP are trained on long-sentence text captions. These captions provide rich context about the objects and surroundings facilitating easy association with the corresponding image. In contrast, the short texts used in image retrieval typically comprise an average of 3-5 words which doesn't match the texts used in training which leads to poor image

retrieval performance. This section described the four modifications we use to improve the performance of image retrieval systems built using pre-trained multi-modal models.

3.1. Fine-Tuning

Fine-tuning is a technique that involves training a model on a specific dataset or task, which allows it to learn and adjust its parameters. Models like CLIP and SigLIP are trained on long text data so when given short text inputs struggle to extract meaningful representations to use in image retrieval. To improve the model's ability to represent both short texts and images effectively, the model is fine-tuned using a dataset of images and short text captions. This fine-tuning process helps enhance the performance of image retrieval systems.

3.2. Expanding Short Texts Using Pre-Trained GPT Model

GPT-2 [10] is a decoder-only transformer architecture that tries to predict the next token based on a previously occurred sequence of tokens. GPT-2 is known for strong performance in zero-shot learning on various NLP tasks, and it can also be fine-tuned to adapt to specific tasks.

For our experiment a pre-trained GPT-2 model (openai-community/gpt2) is used to expand short-text data into longer, more contextual text. The GPT model is fine-tuned to take short text captions as input and output expanded text. The resulting expanded captions are expected to produce more meaningful feature vectors, which could potentially enhance the performance on various downstream tasks.

For fine-tuning we unfroze the last 8 layers of the GPT-2 model to allow weight updates. The loss function used for this was cross-entropy loss which was computed between the model's predictions and the target tokens. AdamW [28] is used as the optimiser. The dataset used is a version of the Flikr30k dataset [8] for which short captions have been generated (as described in Section 4.1). Stopwords are removed from these short captions and pairs of short captions after and before stopword removal are used to fine tune the model.

Once fine tuning is complete input texts (short captions) first go through the fine tuned GPT-2 model, and transform into a longer text before going through the CLIP/SigLIP text encoder.

3.3. SILC Framework

Vision-Language Pre-trained Models (VLP) like CLIP and SigLIP have primarily been trained for image-level tasks such as classification and retrieval. These models excel at open vocabulary tasks at the image level, using contrastive learning to match similar image and text embeddings. However, they struggle with pixel-level tasks like segmentation and object detection. Tasks like object detection and segmentation require an understanding of the local features and also have the ability to make predictions at the pixel level. The SILC Framework aims to enhance models like CLIP and SigLIP by incorporating capabilities for pixel-level tasks.

The SILC Framework combines image-text contrastive pretraining with local-to-global consistency learning by self-distillation. Local-to-global consistency learning involves ensuring local features from cropped patches are consistent with global features from the entire image. This helps models learn strong visual features that enable better local understanding. This improved local understanding not only improves the performance of pixel-level prediction tasks but also for tasks like retrieval because of better local feature understanding.

The SILC Framework consists of a two-tower transformer model, which maintains teacher and student models. The teacher model gets the global view and the student model gets the local view as shown in Figure 1. Here the teacher model is the Exponential Moving Average (EMA) of the student model. This gives a stable base for the student model to learn rather than having a gradient update. The task is to match teachers' feature embedding with only locally cropped images. SILC utilizes 2 global views which are randomly cropped to sizes between 0.4 and 1.0 of the original image and 8 local views which are of size 0.05 to 0.4 of the original image.

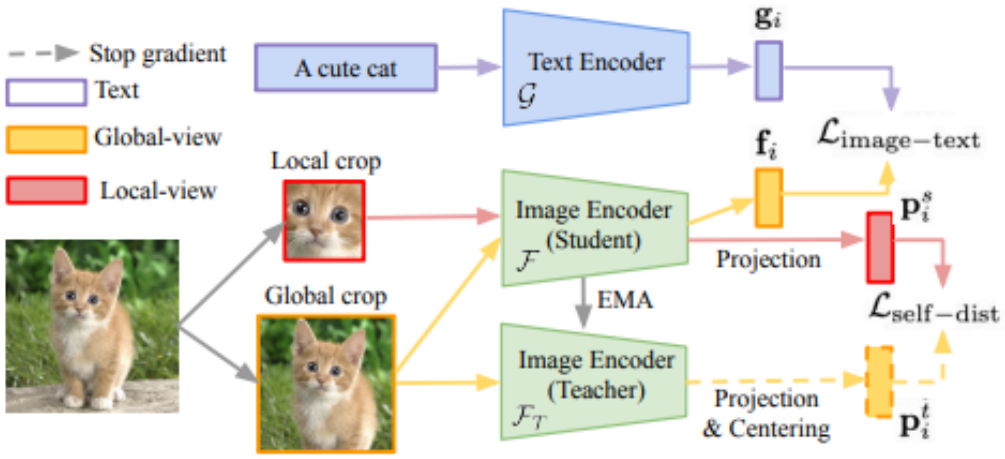

Figure 1: An illustration of the SILC framework (reproduced from [11])

The SILC Framework uses a dual-loss function which combines contrastive loss and self-distillation loss. The first step involves calculating contrastive loss using both global views of the image paired with their associated text. The average loss is taken across views. For the self-distillation component, 16 global-local image pairs are constructed and loss is calculated as the cross-entropy between the probability distributions of the teacher and student models, encouraging consistency between global and local feature representations. The final loss function is the summation of the two components which maintains strong image-text associations while enhancing local feature understanding

3.4. Prompting

Prompt engineering has emerged as an effective way to help Large Language Models produce more accurate and relevant results [29]. This is done by adding relevant background information and instructions to the the prompts which the model uses to generate output.

CLIP uses prompt engineering to handle unlabeled classes in image classification tasks. It uses prompt templates to bridge the gap between textual and visual information, using templates like “A photo of a label”, where label can be replaced by classes from the classification problem.

The short texts in the dataset used in our experiments (see Section 4.1) have an average length of just 3 words. It predominantly contains phrases combining adjectives with object classes (e.g., ’grass green’) or multiple object classes with descriptors (e.g., ’two dogs on road’). To improve the context of short text and bridge the gap between textual and visual data, we wrap the short text data in a prompt template and then pass it through the text encoder.

4. Experimental Setup

This section describes the setup of an experiment designed to evaluate the gap between retrieval performance when long texts and short texts are used, and the effectiveness of the four modifications described in the previous section to reduce this gap. The experiment also compares the effectiveness of systems built using the CLIP and SigLIP models.

4.1. Dataset

Our experiment compares the performance of image retrieval systems using long text captions versus short text captions. This necessitates the need of a dataset with images and two distinct caption sets: long and short. We selected the Flickr30k dataset, which contains 31,783 images and 158,915 captions (5 captions per image). For our purposes, we chose one caption per image from Flickr30k to serve as the

Figure 2: Sample images from the dataset showing long and short captions associated with images

long caption data. As short captions did not exist, we generated them using a pre-trained BLIP model [12].

The captions generated using the BLIP model still exceeded our desired length for short search phrases. The captions generated are further trimmed using stop word removal (which eliminates common words such as articles, prepositions, and conjunctions, which typically don't carry significant meaning in search queries). This approach allows us to directly compare the effectiveness of long and short captions for the same set of images.

The long captions are taken directly from the Flickr30k dataset, averaging 13 words in length. They are descriptive and contain multiple objects, scenes, and background details. They don't solely focus on the main elements dominating the image but also include finer details. In contrast, our generated short text captions average just 3 words. Since it is generated using an image captioning technique, it tends to contain the main focus of the image. For instance, a short caption like "soccer goal" describes an image primarily featuring a soccer ball and goal, while its corresponding long caption, "A goalie is catching a soccer ball", adds more details by specifying the action occurring in the image. Figure 2 illustrates examples of images and their short and long captions.

4.2. Baseline Model

Our experiment uses CLIP and SigLIP models as the baseline. For our CLIP baseline, we employed the CLIP (ViT-B/32) model, which uses a Vision Transformer (ViT) with a 32x32 pixel patch size as its image encoder. The text encoder is a CLIP-based text encoder. This model's architecture consists of 12 transformer layers, 512 embedding dimensions, and 8 attention heads. To adapt it to our dataset, we fine-tuned the last two layers of the CLIP model.

For the SigLIP model, we have used SigLIP (siglip-base-patch16-256) which is a base SigLIP model with a 16 X 16 pixel patch size. It uses a transformer-based text encoder and a Vision Transformer for the image encoder. It has 24 transformer layers, 1024 embedding dimensions, and 16 attention heads. For, the SigLIIP model we have trained all layers (in total 203,202,050 trainable parameters).

4.3. Evaluation Metrics

The evaluation metric used for the Embedding based image retrieval system is Recall@k. Recall@k measures a system's ability to retrieve the relevant items in its top k results.

In our experiments we measure Recall@1, Recall@5, and Recall@10 which is a standard practice. Recall@1 measures a system's ability to retrieve the single most relevant item, while Recall@10 measures the system's ability to at least retrieve relevant items.

4.4. Implementation

We utilized the CLIP model with its default architecture. This includes a transformer-based text encoder and a Vision Transformer (ViT) [30] image encoder. Specifically, we used the pre-trained CLIP model "openai/clip-vit-base-patch32" available through the Hugging Face Transformers library [31]. We used pre-trained weights for most of the model, fine-tuning only the last two layers of both encoders. Our data preprocessing involved loading images using the Python Imaging Library (PIL) [32] and converting them to RGB format. For optimization, we have used the AdamW optimizer with a learning rate of 1e-5 and a weight decay of 0.01. To address the problem of exploding gradients, we implemented gradient clipping with a maximum norm of 0.5. Throughout our experiments, we used a batch size of 16.

For SigLIP, we used the base SigLIP model, specifically the "google/siglip-base-patch16-224" model available through the Hugging Face Transformers package. The image encoder is a Vision Transformer with a patch size of 16, designed to process input images of 256x256 pixels. The text encoder is the default transformer-based architecture provided by SigLIP. Our data preprocessing pipeline involved loading images using the Python Imaging Library (PIL) and converting them to RGB format. These images were then processed using the AutoImageProcessor specific to the SigLIP model. For text data, we employed the AutoTokenizer for tokenization, with padding set to the maximum length. We use the AdamW optimizer with a learning rate of 1e-5 and a weight decay of 0.01. We kept the batch size at 16, aligning with our earlier experiments.

5. Results & Discussion

Table 1 shows the results of our experiment, showing the outcomes of 14 distinct experiments conducted on CLIP and SigLIP models. We evaluate each model configuration using three key performance metrics: Recall@1, Recall@5, and Recall@10 and highlight best performance for each measure.

When starting the experiments, it made more sense to use CLIP in the zero-shot setting rather than fine-tuning the model. Firstly, CLIP is an enormous model with a vast number of parameters, making it computationally expensive to train from scratch. Second, CLIP is highly data-sensitive, requiring massive amounts of data to achieve good generalization. Our dataset, consisting of only 30,000 image-text pairs, was far too small to effectively train CLIP. Given these constraints, we decided to start with the zero-shot CLIP model, leveraging its pre-trained capabilities without further adjustments.

Table 1
Recall@k scores for different models with and without modifications applied. Best performances in each column are highlighted in light blue.

Experiment / Model	Long Text			Short Text		
	Recall@			Recall@		
	1	5	10	1	5	10
CLIP (No Fine-Tuning)	0.547	0.795	0.885	0.257	0.501	0.611
CLIP (No Fine-Tuning) with Prompting	–	–	–	0.281	0.519	0.609
CLIP (No Fine-Tuning) with GPT-2	–	–	–	0.235	0.465	0.577
CLIP (Fine-Tuned)	0.639	0.890	0.940	0.331	0.647	0.766
CLIP (Fine-Tuned) with Prompting	–	–	–	0.257	0.509	0.632
CLIP (Fine-Tuned) with GPT-2	–	–	–	0.316	0.613	0.746
CLIP (Fine-Tuned) with SILC	–	–	–	0.319	0.663	0.773
SigLIP (Fine-Tuned)	0.713	0.913	0.961	0.323	0.659	0.791
SigLIP (Fine-Tuned) with Prompting	–	–	–	0.272	0.548	0.671
SigLIP (Fine-Tuned) with GPT-2	–	–	–	0.362	0.681	0.810
SigLIP (Fine-Tuned) with SILC	–	–	–	0.367	0.683	0.805

At the start we anticipated that CLIP without fine-tuning would perform significantly better with long Flickr captions compared to short ones. This was based on CLIP's pre-training on the WebImageText (WIT) dataset [7], which primarily contains long text captions. As expected we observed a substantial performance gap when long and short search strings were used. CLIP with long search strings achieved a Recall@1 of 0.5470, Recall@5 of 0.7950, and Recall@10 of 0.8850. These scores align closely with those reported by [33], who achieved a Recall@1 of 0.5496 when testing the same base CLIP model (CLIP-ViT-224/32) on the Flickr30k 1k dataset in a zero-shot setting. To improve the performance on shorter search queries, we decided to use methods that would add more context while keeping the solutions simple and avoid making changes to CLIP's architecture. Ultimately, we focused on two approaches that seemed most practical and aligned with our constraints standard prompting and expanding short text using GPT.

5.1. Expanding Queries

When we applied these methods to CLIP in zero-shot settings, neither showed significant improvement over the original model when used with short search strings. The one with prompting model achieved Recall@5 of 0.5190 when short search strings were used, while expanding short search strings with a GPT led to Recall@5 of just 0.4649. The findings suggest that to achieve better representation and improved performance, CLIP needs to be fine-tuned. Fine-tuning will allow the CLIP model to adapt to short texts, different form the texts it was trained with.

For fine-tuning, we determined it wasn't feasible to unfreeze too many layers. Instead, we opted to unfreeze the last two layers of both the text and image encoders in the CLIP model. After applying fine-tuning, there was a significant increase in the performance of the CLIP model when short search strings were used, with the Recall@5 score increasing to 0.6470. Similarly after fine tuning, the scores when short search strings were expanded using a GPT model also increased the Recall@5 to 0.6130. Even after fine tuning prompting didn't improve performance. The fine tuned CLIP model only showed minimal performance increase for long search strings, which indicates what a good zero-shot learner CLIP is.

5.2. Modifying Embedding Spaces

The last method we tried, which gave the best results, was using the SILC framework. This method gives the CLIP model pixel-level prediction ability which improves local feature understanding. When the SILC modification was used the fine-tuned CLIP model achieved Recall@5 of 0.6630.

Based on the impact it had on the CLIP model we used fine tuning on the SigLIP model in all experiments. When used with long search strings the fine-tuned SigLIP model achieved Recall@5 of 0.9130, quite an improvement over the CLIP model. When short search strings were used the fine-tuned SigLIP model outperformed the the equivalent CLIP models in all cases. Overall SigLIP with the SILC Framework modification achieved the best performance for short search strings.

5.3. Discussion

We can conclude that SigLIP is a superior model than CLIP, performing better with both short and long search strings. For short search strings specifically, SigLIP performs best when enhanced with either the SILC framework or GPT-based text expansion. These findings suggest that SigLIP, combined with one of these techniques, represents the current best approach for image retrieval systems, particularly those dealing with short text inputs

A significant limitation of our project was the quality of the short text data used for training and testing the models. This data, generated using the BLIP captioning method, often failed to capture the most relevant aspects of the images. The captions tended to focus on elements occupying the most space in the image, rather than the most important subjects. For instance, in an image with a small object against a large sky background, the caption might simply state "sky is blue", ignoring the main subject. Similarly, when a water body was present in the background with a subject in the foreground, the generated caption might only mention the water, completely omitting the primary subject.

Another issue it has caused is the repetitive values of many of the captions. Captions like "sky blue", "group people", and "water calm" are heavily repeated within the dataset. This could have caused problems during the retrieval stage leading to low Recall@1 scores for all models when short search strings are used.

6. Conclusions & Future Work

In this study, we successfully developed an image retrieval system capable of processing short text queries, addressing a critical gap between the sentence-length data typically used in training and the brief, keyword-based queries common in real-world applications. There was a significant performance gap between long and short text in multi-modal models like CLIP and SigLIP. To narrow this gap, we proposed and evaluated several modifications, with the SILC framework and GPT-based text expansion proving most effective in enhancing short text performance. Surprisingly, simple prompting techniques showed limited benefits. Our experiments consistently demonstrated SigLIP's superiority over CLIP for both long and short text queries.

Our research into methods to improve image retrieval performance when short text queries are used, uncovered several promising approaches. ViSTA [34] and ROSITA [35] utilize scene graphs to enhance contextual information. However, these methods require architectural modifications to CLIP, making

them less suitable for our current purposes. We also explored CoCoOP [36], which replaces standard prompts like "a photo of" with a neural network. This network is trained using backpropagation, and the learned weights represent a prompt added before the original text. Unfortunately, this method also necessitates changes to the model's architecture. In the future, we aim to integrate these methods into our models to assess their impact on short text performance. Additionally, we plan to use more accurate short text data that better represents user queries for image retrieval while reducing repetitive values.

Acknowledgments

This work was supported by Science Foundation Ireland under Grant $12/RC/2289\_P2$.

References

[1] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146 (2018).
[2] J. Lu, D. Batra, D. Parikh, S. Lee, Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, Advances in neural information processing systems 32 (2019).
[3] H. Tan, M. Bansal, Lxmert: Learning cross-modality encoder representations from transformers, arXiv preprint arXiv:1908.07490 (2019).
[4] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, J. Liu, Uniter: Universal image-text representation learning, in: European conference on computer vision, Springer, 2020, pp. 104–120.
[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.
[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.
[7] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, M. Najork, Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning, in: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021, pp. 2443–2449.
[8] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, S. Lazebnik, Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2641–2649.
[9] X. Zhai, B. Mustafa, A. Kolesnikov, L. Beyer, Sigmoid loss for language image pre-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11975–11986.
[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.
[11] M. F. Naeem, Y. Xian, X. Zhai, L. Hoyer, L. Van Gool, F. Tombari, Silc: Improving vision language pretraining with self-distillation, arXiv preprint arXiv:2310.13355 (2023).
[12] J. Li, D. Li, C. Xiong, S. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in: International conference on machine learning, PMLR, 2022, pp. 12888–12900.
[13] K. Simonyan, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[15] A. Vaswani, Attention is all you need, Advances in Neural Information Processing Systems (2017).
[16] J. Devlin, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[17] A. Radford, Improving language understanding by generative pre-training (2018).

[18] F.-L. Chen, D.-Z. Zhang, M.-L. Han, X.-Y. Chen, J. Shi, S. Xu, B. Xu, Vlp: A survey on vision-language pre-training, Machine Intelligence Research 20 (2023) 38–56.
[19] Z. Huang, Z. Zeng, B. Liu, D. Fu, J. Fu, Pixel-bert: Aligning image pixels with text by deep multi-modal transformers, arXiv preprint arXiv:2004.00849 (2020).
[20] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, J. Dai, Vl-bert: Pre-training of generic visual-linguistic representations, arXiv preprint arXiv:1908.08530 (2019).
[21] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: International conference on machine learning, PMLR, 2021, pp. 4904–4916.
[22] K.-H. Lee, X. Chen, G. Hua, H. Hu, X. He, Stacked cross attention for image-text matching, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 201–216.
[23] K. Li, Y. Zhang, K. Li, Y. Li, Y. Fu, Visual semantic reasoning for image-text matching, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 4654–4662.
[24] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE transactions on pattern analysis and machine intelligence 39 (2016) 1137–1149.
[25] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, C. Xu, Filip: Fine-grained interactive language-image pre-training, arXiv preprint arXiv:2111.07783 (2021).
[26] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, et al., Florence: A new foundation model for computer vision, arXiv preprint arXiv:2111.11432 (2021).
[27] J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in: International conference on machine learning, PMLR, 2023, pp. 19730–19742.
[28] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).
[29] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, A. Chadha, A systematic survey of prompt engineering in large language models: Techniques and applications, arXiv preprint arXiv:2402.07927 (2024).
[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[31] T. Wolf, Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).
[32] A. Clark, Pillow (pil fork) documentation, 2015. URL: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.
[33] Z.-Y. Dou, Y. Xu, Z. Gan, J. Wang, S. Wang, L. Wang, C. Zhu, P. Zhang, L. Yuan, N. Peng, et al., An empirical study of training end-to-end vision-and-language transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18166–18176.
[34] M. Cheng, Y. Sun, L. Wang, X. Zhu, K. Yao, J. Chen, G. Song, J. Han, J. Liu, E. Ding, et al., Vista: Vision and scene text aggregation for cross-modal retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5184–5193.
[35] Y. Cui, Z. Yu, C. Wang, Z. Zhao, J. Zhang, M. Wang, J. Yu, Rosita: Enhancing vision-and-language semantic alignments via cross-and intra-modal knowledge integration, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 797–806.
[36] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16816–16825.