VEMO: A Versatile Elastic Multi-Modal Model for Search-Oriented Multi-Task Learning

Nanyi Fei[1†[0000−0002−3852−9298]], Hao Jiang[2†], Haoyu Lu[3], Jinqiang Long[3], Yanqi Dai[3], Tuo Fan[2], Zhao Cao[2], and Zhiwu Lu[3∗[0000−0003−0280−7724]]

[1] School of Information, Renmin University of China, Beijing, China
feinanyi@ruc.edu.cn
[2] Huawei Poisson Lab, Hangzhou, Zhejiang, China
{jianghao66, fantuo1, caozhao1}@huawei.com
[3] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{lhy1998, longjinqiang, yanqidai, luzhiwu}@ruc.edu.cn

Abstract. Cross-modal search is one fundamental task in multi-modal learning, but there is hardly any work that aims to solve multiple cross-modal search tasks at once. In this work, we propose a novel **V**ersatile **E**lastic **M**ulti-m**O**dal (VEMO) model for search-oriented multi-task learning. VEMO is versatile because we integrate cross-modal semantic search, named entity recognition, and scene text spotting into a unified framework, where the latter two can be further adapted to entity- and character-based image search tasks. VEMO is also elastic because we can freely assemble sub-modules of our flexible network architecture for corresponding tasks. Moreover, to give more choices on the effect-efficiency trade-off when performing cross-modal semantic search, we place multiple encoder exits. Experimental results show the effectiveness of our VEMO with only 37.6% network parameters compared to those needed for uni-task training. Further evaluations on entity- and character-based image search tasks also validate the superiority of search-oriented multi-task learning.

Keywords: multi-modal model · multi-task learning · cross-modal search.

1 Introduction

Cross-modal search [41, 34, 19, 30, 47] is fundamental in multi-modal learning. Humans obviously possess cross-modal search ability. We can not only find images with proper descriptions (*i.e.*, cross-modal semantic search), but also match images with given entities (*i.e.*, entity-based image search), as well as find images having the given text in them (*i.e.*, character-based image search). These different types of cross-modal search tasks are certainly in need in real scenarios.

However, researches on multi-modal search models in the literature mostly focus on only one type of search task. In recent years, techniques towards cross-modal semantic search (also known as cross-modal retrieval) [11, 33, 43, 16, 27, 15]

[∗] Z. Lu is the corresponding author. [†] N. Fei and H. Jiang contribute equally.

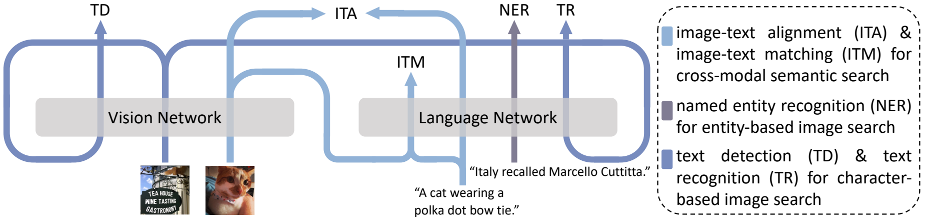

TD
ITA
NER
TR
ITM
Vision Network
Language Network
"Italy recalled Marcello Cuttitta."
"A cat wearing a polka dot bow tie."
■ image-text alignment (ITA) & image-text matching (ITM) for cross-modal semantic search
■ named entity recognition (NER) for entity-based image search
■ text detection (TD) & text recognition (TR) for character-based image search

Fig. 1: Overview of our VEMO. It integrates cross-modal semantic search (including ITA & ITM), named entity recognition, and scene text spotting (including TD & TR) for search-oriented multi-task learning.

have achieved great success. Significant progress has also been made in the fields of named entity recognition [17, 46, 44, 51] and scene text spotting/optical character recognition [28, 8, 42, 49], which are closely related to entity- and character-based image search tasks, respectively. Unfortunately, there is hardly any work that aims to solve multiple cross-modal search tasks at once.

To fill the void on search-oriented multi-task learning, we propose a **V**ersatile **E**lastic **M**ulti-m**O**dal model (VEMO), which integrates cross-modal semantic search (CMSS), named entity recognition (NER), and scene text spotting (STS) in a unified framework. The overview of our VEMO is presented in Fig. 1. Once trained, besides these three explicitly learned tasks, VEMO can go beyond and carry out entity- and character-based image search tasks with the help of NER and STS, respectively. We devise a flexible model architecture that allows the simultaneous training of these tasks. That is, we modify ViT [7] and BERT [6] as our two main networks, from which we can take out sub-modules (encoders and decoders) for corresponding tasks. Specifically, for CMSS, we need an image encoder and a text encoder for instance-level image-text alignment, and we also need a multi-modal encoder for finer-grained token-level image-text matching. For NER, we simply need a text encoder for token classification. For STS, we need an image encoder and two decoders: a location decoder for text detection and a text decoder for text recognition. All these encoders and decoders come from our modified ViT and BERT, with parameter re-use not only among tasks but also within one task. Our design keeps VEMO in a relatively small scale and also makes VEMO very flexible because we can choose modules as we need.

For the inference of CMSS, the instance-level image-text alignment is efficient because we only need to extract the query embedding and compute similarities with all pre-extracted candidate embeddings at the instance level. On the other hand, although the token-level image-text matching is very time-consuming, it is more effective because of the finer-grained modeling. To give more choices between the two extremes, we place multiple exits at different layers of the multi-modal encoder when performing image-text matching. In this way, we can freely adjust the effect-efficiency trade-off according to the usage scenarios.

Our contributions are summarized here: (1) We propose a novel **V**ersatile **E**lastic **M**ulti-m**O**dal (VEMO) model for search-oriented multi-task learning. We integrate CMSS, NER, and STS into a unified framework, where the latter two can further contribute to entity- and character-based image search tasks, respectively, indicating the versatility of VEMO. (2) We design a flexible model architecture, where we can use different modules for corresponding tasks, with parameter re-use among these modules. Moreover, we introduce selectable image-text matching exits, making VEMO more elastic. (3) Experiments show that VEMO saves a lot of parameters while achieving comparative results with independently trained uni-task models. Further evaluations on entity- and character-based image search tasks also demonstrate the effectiveness of our search-oriented model.

2 Related Work

Multi-Modal Multi-Task Search has drawn very little attention in the literature. The only related work we find is Multi-task Unified Model (MUM) [30] from Google, which is devised for their new search function: multisearch [47]. Multisearch allows the input of image and text together as search query, and aims to understand them in more natural ways to form a composed query. It is different from our VEMO in that MUM focuses on performing end-to-end search via one model to finally replace all the steps of search engines, while we simply design a multi-modal multi-task model for different types of search tasks.

Multi-Task Learning (MTL) [3, 50, 40] aims to leverage useful knowledge in multiple related tasks to improve the model generalization ability. Most existing MTL studies in computer vision [13, 23, 22, 29] and natural language processing [35, 39, 3] focus on tasks that can be summarized into a uniform format, making the design of multi-task model neat and orderly (*e.g.*, one shared backbone with multiple task-specific heads). However, we consider tasks that differ in formalizations, thus making our model architecture much more complex.

Cross-Modal Semantic Search is also known as cross-modal retrieval. Recent works can be generally grouped into three types. Dual-stream methods [1, 45, 38, 27] adopt separate vision and language encoders for global instance-level image-text alignment, which are efficient during evaluation. Single-stream ones [4, 10, 48, 16] resort to one multi-modal encoder, allowing finer-grained modality interaction, which are thus more effective but time-consuming. Methods adopting hybrid architectures [15, 43] integrate dual-stream and single-stream, offering both choices. Inspired by BLIP [15], we also adopt the hybrid architecture and extend it to more tasks. Notably, with multiple exits of the multi-modal encoder for token-level image-text matching, our VEMO is even more elastic.

Named Entity Recognition [17, 46, 44, 51] aims to identify named entities in text and classify them into pre-defined categories (*e.g.*, person and location). **Scene Text Spotting** [28, 8, 42, 49] aims to detect and recognize text in natural scenes. To our best knowledge, we are the first to combine these two tasks with cross-modal semantic search for search-oriented multi-task learning.

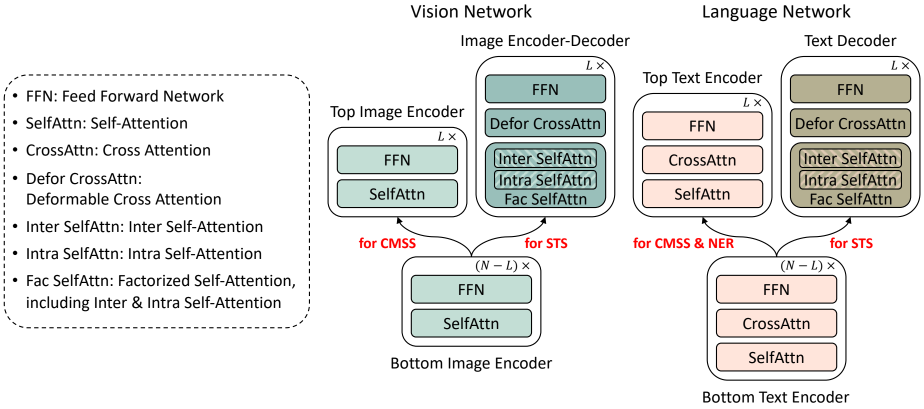


Fig. 2: The overall architecture of VEMO. Parameter re-use among different search-oriented tasks makes VEMO versatile, elastic, and meanwhile storage efficient.

3 Methodology

3.1 Framework Overview

The proposed VEMO is an end-to-end framework that handles cross-modal semantic search (CMSS), named entity recognition (NER), and scene text spotting (STS) in a unified manner. As shown in Fig. 2, VEMO contains two main networks of N layers: a vision network modified from ViT [7] and a language network modified from BERT [6]. Each network can be divided into three parts: bottom $(N-L)$ layers only work as encoding layers, while top L layers have two branches. In Sec. 3.2, we describe the CMSS task, where the bottom and top image encoders as well as the bottom and top text encoders are used. In Sec. 3.3, we introduce NER, where only part of the bottom and top text encoders are used. Then we describe STS in Sec. 3.4, which contains text detection and recognition as two sub-tasks. The bottom image encoder and part of the image encoder-decoder are used for encoding input images, followed by text detection using the image encoder-decoder and text recognition using the text decoder.

3.2 Cross-Modal Semantic Search (CMSS)

Given a query from one modality, CMSS aims to search samples from another modality that semantically match the query. As illustrated in Fig. 3(a), we consider two training losses for modality interactions at two levels: a cross-modal momentum contrastive loss for instance-level image-text alignment, and an token-level image-text matching loss for finer-grained modality interaction.

Cross-Modal Momentum Contrastive (CMMC) Loss. The CMMC loss is adopted to project samples from different modalities into a unified embedding space for image-text alignment at the global instance level. Here, the total N

layers of the top and bottom image (or text) encoders are used as the uni-modal image (or text) encoder, which encodes image patches (or word tokens) into a sequence of features along with an additional [CLS-I] (or [CLS-T]) token representing the global sample feature. Note that the cross attention (CrossAttn) module in each layer is skipped for the uni-modal text encoder.

For each image-text pair (I_i, T_i) in a batch $\mathcal{B}$ sampled from the CMSS dataset, let $\mathbf{f}_i^I$ and $\mathbf{f}_i^T$ denote the global image and text embeddings after linear projection on the [CLS-I] and [CLS-T] features, respectively. Inspired by uni-modal MoCo [9], we maintain two queues $\mathcal{Q}^I$ and $\mathcal{Q}^T$ to keep the most recent N_q image-text features obtained from the momentum uni-modal encoders. For each I_i, we regard the momentum feature $\hat{\mathbf{f}}_i^T$ of its paired T_i as the positive sample, and take all samples in $\mathcal{Q}^T$ as negatives. The image-to-text contrastive loss is:

$$\mathcal{L}_{\text{i2t}} = -\frac{1}{|\mathcal{B}|} \sum\nolimits_{(I_i,T_i)\in\mathcal{B}} \log \frac{\text{pos}(\mathbf{f}_i^I, \hat{\mathbf{f}}_i^T, \tau)}{\text{pos}(\mathbf{f}_i^I, \hat{\mathbf{f}}_i^T, \tau) + \text{neg}(\mathbf{f}_i^I, \mathcal{Q}^T, \tau)}, \tag{1}$$

where τ is the temperature parameter, and

$$\text{pos}(\mathbf{f}_i^I, \hat{\mathbf{f}}_i^T, \tau) = \exp(\mathbf{f}_i^I \cdot \hat{\mathbf{f}}_i^T/\tau), \quad \text{neg}(\mathbf{f}_i^I, \mathcal{Q}^T, \tau) = \sum\nolimits_{\mathbf{q}_j^T\in\mathcal{Q}^T} \exp(\mathbf{f}_i^I \cdot \mathbf{q}_j^T/\tau). \tag{2}$$

Similarly, the text-to-image contrastive loss is defined as:

$$\mathcal{L}_{\text{t2i}} = -\frac{1}{|\mathcal{B}|} \sum\nolimits_{(I_i,T_i)\in\mathcal{B}} \log \frac{\text{pos}(\mathbf{f}_i^T, \hat{\mathbf{f}}_i^I, \tau)}{\text{pos}(\mathbf{f}_i^T, \hat{\mathbf{f}}_i^I, \tau) + \text{neg}(\mathbf{f}_i^T, \mathcal{Q}^I, \tau)}, \tag{3}$$

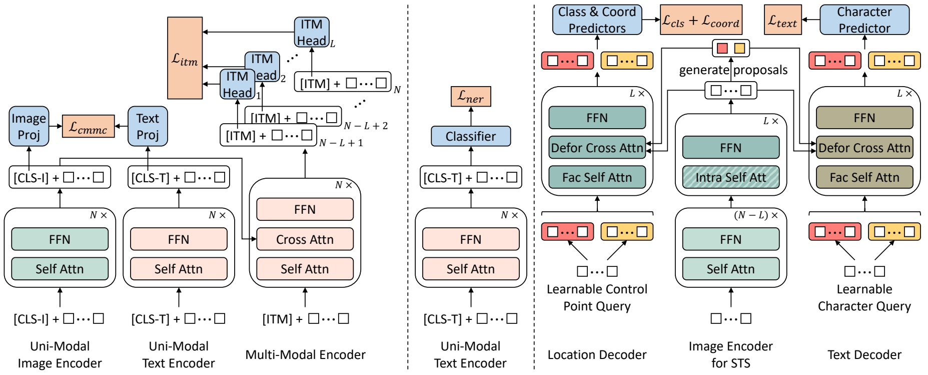
where $\hat{\mathbf{f}}_i^I$ is the momentum feature of I_i. The total CMMC loss is simply as:

$$\mathcal{L}_{\text{cmmc}} = \mathcal{L}_{\text{i2t}} + \mathcal{L}_{\text{t2i}}. \tag{4}$$

After loss calculation, the momentum features of paired images and texts from the current batch are then pushed into the corresponding momentum queues, meanwhile the earliest $|\mathcal{B}|$ samples in both queues are popped out.

Image-Text Matching (ITM) loss. The ITM loss is adopted to model finer modality interactions at the token level. Concretely, we use all N layers of the top and bottom text encoders as the multi-modal encoder. Note that the self-attention (SelfAttn) and the feed forward network (FFN) in each layer share weights with those in the same layer of the uni-modal text encoder, which largely reduces the number of network parameters. The multi-modal encoder takes image-text pairs as input, and performs binary classification to predict whether the input pair is matched. For the raw input text, we append an [ITM] token to represent the multi-modal feature, where image information is encoded by inputting the final sequence of patch features from the uni-modal image encoder as the "keys" and "values" for the CrossAttn in each layer.

Furthermore, to freely adjust the effect-efficiency trade-off, we allow the multi-modal encoder to exit at each of the top L layers. Specifically, for each



(a) CMSS (b) NER (c) STS

Fig. 3: Schematic illustration of three search-oriented training tasks in VEMO. Note that sub-modules with the same name and color are shared across different tasks and encoders/decoders. Multiple exits for ITM make VEMO more elastic.

input pair (I, T), let $f^{\text{ITM}}_{N-L+l}(I, T)$ denote the `[ITM]` feature at the $(N-L+l)$-th layer $(l = 1, \cdots, L)$. A two-class linear classifier h^{ITM}_l (*i.e.*, the ITM head) is adopted, which is followed by a softmax function σ, resulting in two probabilities $\sigma(h^{\text{ITM}}_l(f^{\text{ITM}}_{N-L+l}(I, T))) \in \mathbb{R}^2$ $(l = 1, \cdots, L)$. Without loss of generality, we regard its first element as the matching score of (I, T), denoted as $\text{ITM}_l(I, T)$.

To calculate the ITM loss, inspired by BLIP [15], for each image I_i (or each text T_i) in batch $\mathcal{B}$, we first sample a hard negative sample $\text{HardNeg}(I_i) \in \{T_j | j = 1, \cdots, |\mathcal{B}|, j \neq i\}$ (or $\text{HardNeg}(T_i) \in \{I_j | j = 1, \cdots, |\mathcal{B}|, j \neq i\}$). In this way, we have one positive pair (I_i, T_i) and two negative pairs $(I_i, \text{HardNeg}(I_i))$ and $(\text{HardNeg}(T_i), T_i)$ for each $(I_i, T_i) \in \mathcal{B}$. In each training iteration, we average the ITM losses calculated over all top L layers of the multi-modal encoder:

$$\begin{aligned}\mathcal{L}_{\text{itm}} =& \frac{1}{3L|\mathcal{B}|} \sum\nolimits_{(I_i,T_i)\in\mathcal{B}} \sum\nolimits_{l\in\{1,\cdots,L\}} [\text{BCE}(1, \text{ITM}_l(I_i, T_i)) \\ &+ \text{BCE}(0, \text{ITM}_l(I_i, \text{HardNeg}(I_i))) + \text{BCE}(0, \text{ITM}_l(\text{HardNeg}(T_i), T_i))],\end{aligned} \tag{5}$$

where $\text{BCE}(y, \hat{y}) = -y \log(\hat{y}) - (1-y)\log(1-\hat{y})$ is the binary cross-entropy.

Finally, the total loss for cross-modal semantic search is:

$$\mathcal{L}_{\text{cmss}} = \mathcal{L}_{\text{cmmc}} + \mathcal{L}_{\text{itm}}. \tag{6}$$

3.3 Named Entity Recognition (NER)

As illustrated in Fig. 3(b), for NER, we use the same uni-modal text encoder as in calculating CMMC loss. We simply adopt a two-layer classifier on top of the text encoder and compute the token classification loss. Specifically, for each text T in a batch $\mathcal{B}_{\text{ner}}$ sampled from the NER dataset, we can obtain a score matrix $\hat{\mathbf{S}} \in \mathbb{R}^{N_{\text{seq}} \times N_{\text{c}}}$, where N_{seq} and N_{c} are the token sequence length and the number of classes, respectively. The NER loss is then defined as:

$$\mathcal{L}_{\mathrm{ner}} = \frac{1}{|\mathcal{B}_{\mathrm{ner}}|} \sum\nolimits_{T \in \mathcal{B}_{\mathrm{ner}}} \mathrm{CE}(\mathbf{S}, \hat{\mathbf{S}}), \tag{7}$$

where $\mathbf{S} \in \mathbb{R}^{N_{\mathrm{seq}} \times N_{\mathrm{c}}}$ is the one-hot ground-truth label matrix over N_{seq} tokens of sample T, and $\mathrm{CE}(\cdot, \cdot)$ is the batched cross-entropy function over N_{seq} tokens.

3.4 Scene Text Spotting (STS)

Inspired by TESTR [49], we design an end-to-end STS approach involving one image encoder and two decoders. As illustrated in Fig. 3(c), the architectures of both decoders are the same, *i.e.*, each decoder layer contains a factorized SelfAttn (including an intra SelfAttn and an inter SelfAttn), a deformable CrossAttn [52], and an FFN. As For the image encoder, to minimize the possible learning conflict of fully sharing it with the uni-modal image encoder used in CMSS, we only reuse the bottom $(N - L)$ layers. The top L layers of the image encoder for STS come from the location decoder (that is why we also call it the image encoder-decoder), where only the intra SelfAttn and FFN are used in each layer.

For each image I in an STS batch $\mathcal{B}_{\mathrm{sts}}$, the encoder first extracts the sequence of image patch features. Then for each patch, we predict a coarse bounding box (*i.e.*, proposal) and a probability of having text within the box. Only proposals with top-P probability values are selected for further location and text decoding.

Location Decoder Loss. To localize text in arbitrary shapes, we expect N_{ctrl} control points to enclose a polygon. Specifically, we adopt N_{ctrl} learnable control point query tokens $\mathbf{C} \in \mathbb{R}^{N_{\mathrm{ctrl}} \times d_{\mathrm{ctrl}}}$, where d_{ctrl} is the dimension of each control point token embedding. To help the location decoding process, we embed the proposal information into $\mathbf{C}$ by first making P copies of $\mathbf{C}$ and then adding the transformed p-th proposal into the p-th copy. In this way, we obtain the input of the location decoder as $\mathcal{C} = \{\mathbf{C}^{(p)} \in \mathbb{R}^{N_{\mathrm{ctrl}} \times d_{\mathrm{ctrl}}}\}_{p=1}^{P}$, where each group focuses on one region in the image. In each layer, an intra SelfAttn is first performed over N_{ctrl} query tokens for each of the P groups independently, then an inter SelfAttn is performed over P tokens for the same j-th $(j = 1, \cdots, N_{\mathrm{ctrl}})$ control point. In the following deformable CrossAttn, P proposals are naturally used as reference points to sample "keys" from the image patch features (see details of the deformable attention in [52]). After obtaining the output query embeddings from the location decoder, for each control point group, we devise a binary classifier to predict whether this region has text in it and a coordinate predictor to predict the coordinates of N_{ctrl} control points. Since not all P groups' corresponding image regions contain text, an injective function $\phi : \{1, \cdots, N_{\mathrm{gt}}\} \to \{1, \cdots, P\}$ is needed, where N_{gt} is the number of ground-truth text annotations in an image. This bipartite matching problem can be efficiently solved by the Hungarian algorithm [14]. Let $s^{(p)}$ denote the classification probability of the p-th control point group and $\mathit{\Phi} = \{\phi(g)\}_{g=1}^{N_{\mathrm{gt}}} \subseteq \{1, \cdots, P\}$ denote the index set of groups containing text. The classification loss is then defined as a focal loss [20]:

$$\mathcal{L}_{\text{cls}}^{\text{dec}} = -\sum\nolimits_{p\in\Phi} \alpha(1-s^{(p)})^{\gamma} \log s^{(p)} - \sum\nolimits_{p\notin\Phi} (1-\alpha)(s^{(p)})^{\gamma} \log(1-s^{(p)}), \quad (8)$$

where α and γ are two hyper-parameters. Let $\mathbf{z}^{(g)} \in \mathbb{R}^{N_{\text{ctrl}}\times 2}$ $(g = 1, \cdots, N_{\text{gt}})$ denote the g-th ground-truth annotation in an image and $\hat{\mathbf{z}}^{(p)} \in \mathbb{R}^{N_{\text{ctrl}}\times 2}$ $(p = 1, \cdots, P)$ denote the predicted control point coordinates of the p-th query group. We define the coordinate regression loss as:

$$\mathcal{L}_{\text{coord}} = \sum\nolimits_{g\in\{1,\cdots,N_{\text{gt}}\}} \|\mathbf{z}^{(g)} - \hat{\mathbf{z}}^{(\phi(g))}\|. \quad (9)$$

Text Decoder Loss. Like the learnable control point query tokens for location decoding, we adopt N_{char} learnable character query tokens for the text decoder. Similarly, the query is duplicated into P copies before being input into the text decoder. By adopting a character classifier, we can finally obtain the predicted classification scores $\hat{\mathbf{t}}^{(p)} \in \mathbb{R}^{N_{\text{char}}\times(N_{\text{voc}}+1)}$ for p-th query group, where N_{voc} is the number of characters in the vocabulary and an additional null class is needed because N_{char} is larger than the length of ground-truth text in most cases. The text recognition loss is then defined as the character classification loss:

$$\mathcal{L}_{\text{text}} = \sum\nolimits_{g\in\{1,\cdots,N_{\text{gt}}\}} \text{CE}(\mathbf{t}^{(g)}, \hat{\mathbf{t}}^{(\phi(g))}), \quad (10)$$

where $\mathbf{t}^{(g)} \in \mathbb{R}^{N_{\text{char}}\times(N_{\text{voc}}+1)}$ denotes the one-hot labels over N_{char} characters in the g-th ground-truth text in an image.

Encoder Loss. Since the two decoders both rely on the output of the image encoder (*i.e.*, image patch features and the generated top-P proposals), we introduce extra constraints for the encoder. As we have mentioned, for each image patch feature, we predict a coarse bounding box and a probability of having text within the box. We thus adopt similar binary classification loss $\mathcal{L}_{\text{cls}}^{\text{enc}}$ and bounding box coordinate regression loss $\mathcal{L}_{\text{bbox}}$ like those for the location decoder. An extra generalized IoU loss [36] for bounding box regression $\mathcal{L}_{\text{giou}}$ is also used.

Overall, the final scene text spotting loss is:

$$\begin{aligned}\mathcal{L}_{\text{sts}} = \frac{1}{|\mathcal{B}_{\text{sts}}|} \sum\nolimits_{I\in\mathcal{B}_{\text{sts}}} (&\lambda_{\text{cls}}\mathcal{L}_{\text{cls}}^{\text{dec}} + \lambda_{\text{coord}}\mathcal{L}_{\text{coord}} + \lambda_{\text{text}}\mathcal{L}_{\text{text}} \\ &+ \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}^{\text{enc}} + \lambda_{\text{coord}}\mathcal{L}_{\text{bbox}} + \lambda_{\text{giou}}\mathcal{L}_{\text{giou}}),\end{aligned} \quad (11)$$

where λ_{cls}, λ_{coord}, λ_{text}, and λ_{giou} are all hyper-parameters.

4 Experiments

4.1 Datasets

Our VEMO is trained on five datasets simultaneously: two for cross-modal semantic search (CMSS), one for named entity recognition (NER), and two for scene text spotting (STS).

We select two widely-used datasets for CMSS: (1) **MSCOCO** [21] is an image-text dataset of 123,287 images, with each image annotated by 5 captions. We follow [16, 27, 15] and split the dataset into 113,287 training, 5,000 validation, and 5,000 test images. (2) **Flickr30K** [32] is smaller, with 31,000 images and 158,915 captions in total. As in [16, 27, 15], we split the dataset into 29,000 training, 1,000 validation, and 1,000 test images. For performance evaluation, Recall@k (R@k, $k = 1, 5, 10$) and Recall@Mean (R@Mean) are reported, where R@Mean is the average of R@1, R@5, and R@10.

For NER, we adopt the classic **CoNLL-2003** [37], which has 203,621 training, 51,362 validation, and 46,435 testing tokens. It has four types of entities: persons, organizations, locations, and miscellaneous names. We report precision (P), recall (R), and the F1 score as the evaluation results.

For STS, two popular datasets are selected: (1) **Total-Text** [5] has 1,255 training images and 300 test images, with each image containing curved texts. (2) **ICDAR 2015** [12] contains 1,000 training images and 500 test images. It is more difficult because the images are from hand-held cameras in the wild. The standard evaluation protocols used for these datasets are followed.

4.2 Implementation Details

The vision and language networks of our VEMO are modified from ViT-B/16 [7] and BERT-Base [6], respectively. Thus the number of layers N for each network is 12. And the number of location/text decoder layers L is set to 6.

For CMSS, the input image size is 384×384, and the maximum length of input text is 35. The batch size $|\mathcal{B}| = 240$. The negative queue size $N_q = 57,600$. And τ in Eqs. (1) – (3) is learnable with the initialization of 0.07. For NER, we set $|\mathcal{B}_{\text{ner}}| = 128$, $N_{\text{seq}} = 128$, and $N_{\text{c}} = 12$. For STS, the input image size is $1,024 \times 1,024$. We set $|\mathcal{B}_{\text{sts}}| = 8$, $N_{\text{char}} = 25$, and $N_{\text{ctrl}} = 16$. Top-100 (*i.e.*, $P = 100$) proposals are selected for further location and text decoding. In Eq. (8), $\alpha = 0.25$ and $\gamma = 2.0$. In Eq. (11), $\lambda_{\text{cls}} = 2$, $\lambda_{\text{coord}} = 5$, $\lambda_{\text{text}} = 4$, and $\lambda_{\text{giou}} = 2$.

We employ the AdamW [26] optimizer with the initial learning rate of 1e-5. And we adopt the cosine learning rate scheduler for training a total of 5 epochs.

Before multi-task training, we load the "BLIP w/ ViT-B (14M)" model [15], fix all its parameters, and only train the image encoder-decoder and the text decoder for STS (*i.e.*, STS pre-training). Following TESTR [49], three STS pre-training datasets are used: SynthText 150K [24], MLT 2017 [31], and Total-Text [5] (see details in [49]). The initial pre-training learning rate is 1e-4. We also adopt the cosine learning rate scheduler for training 400K iterations.

4.3 Main Results

We first present the results by our VEMO using different multi-task training strategies, as well as the uni-task training results in Table 1. Specifically, four strategies are used to train our VEMO: (1) "sum" – three task losses are simply

Table 1: Main results (%) on 5 datasets across 3 tasks. Notations: #Params – the number of model parameters; R@Mean – the average of image-to-text and text-to-image R@Mean; Δm – the overall performance gain of each VEMO variant w.r.t. the uni-task method, defined as the difference of $\frac{1}{3}(\frac{1}{2}(\text{R@Mean(MSCOCO)}+\text{R@Mean(Flickr30K)})+\text{F1(CoNLL-2003)}+\frac{1}{2}(\text{F1(Total-Text)}+\text{F1(ICDAR 2015)}))$. Best results in each column are highlighted in bold, and second-best ones are underlined.

		CMSS		NER			STS						
Method	#Params	MSCOCO	Flickr30K	CoNLL-2003			Total-Text			ICDAR 2015			Δm
		R@Mean	R@Mean	P	R	F1	P	R	F1	P	R	F1	
Uni-Task	964.22M	85.16	**96.48**	75.45	83.71	79.37	62.94	**51.26**	56.50	38.31	**26.19**	**31.11**	-
VEMO-sum	362.50M	85.04	96.07	**79.86**	83.39	**81.59**	63.72	50.60	56.41	38.91	24.07	29.74	**0.41**
VEMO-log_sum	362.50M	84.87	96.06	75.04	**84.21**	79.36	**64.95**	**51.26**	**57.30**	**40.91**	24.27	30.46	-0.10
VEMO-DWA	362.50M	84.98	96.14	79.37	83.02	81.15	63.30	50.55	56.21	39.07	23.93	29.68	0.22
VEMO-iterative	362.50M	**85.48**	96.38	75.10	83.45	79.05	63.46	50.99	56.54	40.42	25.28	31.10	-0.07

added together, *i.e.*, $\mathcal{L}_{\text{sum}} = \mathcal{L}_{\text{cmss}} + \mathcal{L}_{\text{ner}} + \mathcal{L}_{\text{sts}}$. (2) "log_sum" – the logarithm of three losses are added, *i.e.*, $\mathcal{L}_{\text{log\_sum}} = \log \mathcal{L}_{\text{cmss}} + \log \mathcal{L}_{\text{ner}} + \log \mathcal{L}_{\text{sts}}$. (3) "DWA" [23] – an adaptive weight is assigned to each loss according to the change rate of the loss value. Generally, if the decrease rate of one loss becomes small, then the assigned weight also becomes small. The final loss $\mathcal{L}_{\text{DWA}} = w_1\mathcal{L}_{\text{cmss}} + w_2\mathcal{L}_{\text{ner}} + w_3\mathcal{L}_{\text{sts}}$, where $w_1 + w_2 + w_3 = 3$. (4) "iterative" – we first calculate loss for one task, and then perform back propagation immediately before calculating other tasks' losses. This process is conducted iteratively among 3 tasks. As for uni-task training, we train one model for each dataset independently, using the same hyper-parameter settings as those in multi-task training.

Besides results on each dataset, we report Δm, the overall performance gain of VEMO models w.r.t. uni-task results. We also give the number of network parameters needed for each method, where for uni-task training, we report the total parameter amount of all uni-task models separately trained on five datasets.

We can observe from Table 1 that although VEMO only needs about 37.6% network parameters of those for uni-task training, VEMO variants achieve comparative results with uni-task training. This clearly validates the effectiveness of VEMO with such great reduction on the number of model parameters. Among four VEMO variants, different multi-task training strategies seem to place emphasis on different tasks, and the simplest "sum" strategy has the best overall performance. It is not surprising because unlike those strongly related tasks in conventional multi-task learning, our three chosen tasks diverse in modalities and formalizations, which leads to a more complicated multi-task balancing scenario.

We further present detailed results on MSCOCO and Flickr30K in Table 2. Note that our VEMO variants are trained in a multi-task manner and they also have multiple image-text matching exits, which clearly make the training more difficult. Despite that, Table 2 shows that VEMO variants generally achieve comparative results with current best, even beating ALIGN with 18B pre-training data. This indicates that search-oriented multi-task learning and placing multiple image-text matching exits do not harm the performance on CMSS.

Table 2: Detailed results (%) on MSCOCO and Flickr30K for CMSS. Notations: # PT Images – the number of images in pre-training data; MTL – multi-task learning; I2T/T2I – image-to-text/text-to-image. Best results in each group are in bold.

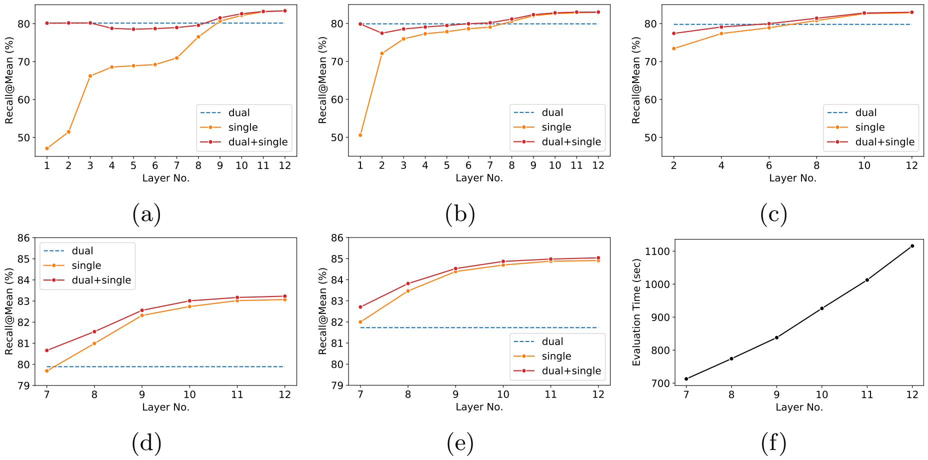
Method	# PT Images	MTL	MSCOCO						Flickr30K					
			I2T Search			T2I Search			I2T Search			T2I Search		
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER-Base [4]	4M	no	64.4	87.4	93.1	50.3	78.5	87.2	85.9	97.1	98.8	72.5	92.4	96.1
OSCAR-Base [18]	4M	no	70.0	91.1	95.5	54.0	80.8	88.5	-	-	-	-	-	-
ALIGN [11]	1.8B	no	77.0	93.5	96.9	59.9	83.3	89.8	95.3	**99.8**	**100.0**	84.9	97.4	98.6
ALBEF [16]	14M	no	77.6	94.3	97.2	60.7	84.3	90.5	95.9	**99.8**	**100.0**	85.6	**97.5**	**98.9**
COTS [27]	14M	no	69.0	90.4	94.9	52.4	79.0	86.9	90.6	98.7	99.7	76.5	93.9	96.6
BLIP-Base [15]	14M	no	**80.6**	**95.2**	**97.6**	**63.1**	**85.3**	**91.1**	**96.6**	**99.8**	**100.0**	**87.2**	**97.5**	98.8
Uni-Task	14M	no	80.60	94.20	96.84	63.53	85.19	90.58	**96.00**	99.70	99.90	**87.08**	**97.48**	**98.70**
VEMO-sum	14M	yes	80.54	94.32	96.88	63.17	84.99	90.34	95.40	99.70	**100.00**	85.66	97.14	98.52
VEMO-log_sum	14M	yes	80.46	94.14	96.64	62.96	84.69	90.32	95.50	99.50	99.90	86.08	97.06	98.34
VEMO-DWA	14M	yes	80.58	94.12	96.72	63.27	84.94	90.27	95.70	99.70	**100.00**	85.86	97.06	98.52
VEMO-iterative	14M	yes	**81.08**	**94.70**	**97.12**	**63.90**	**85.27**	**90.79**	95.90	**99.80**	99.90	86.58	97.42	98.68

4.4 Selectable ITM Exits

To freely adjust the effect-efficiency trade-off for CMSS, we place multiple ITM exits at top 6 layers of the multi-modal encoder (see Fig. 3(a)). Below we give a detailed analysis of why we choose the top 6 layers.

In subfigures (a) – (d) of Fig. 4, we show results by uni-task CMSS training of our VEMO on MSCOCO. The blue dash line shows the instance-level image-text alignment result by only using separate uni-modal encoders (denoted as dual-encoder result). The orange line shows the result by using the multi-modal encoder only (denoted as single-encoder result). And the red line result is the ensemble of dual-encoder and single-encoder.

In Fig. 4(a), the model is trained in two stages. We first train the model with only one ITM head at the last layer. Then we place 11 ITM heads at every previous layer and only train these 11 heads by freezing the trained model at the first stage. We can see that single-encoder results (orange line) generally get better as the layer is closer to the top, but only top 4 layers produce better results than simple dual-encoder. It is also observed that the ensemble performance (red line) first drops and then rises, because as the multi-modal encoder uses more and more layers, the contribution of the dual-encoder decreases. In Fig. 4(b), we train the model with all 12 ITM heads at once, where the performance of 2nd – 8th layers is improved a lot. This can be explained as: in Fig. 4(a), all 1st to 11th layers are only trained to provide information for the subsequent layer, while in Fig. 4(b), these layers are also trained to classify. But still, only the top 5 layers deliver better performance than simple dual-encoder. Another interesting phenomenon is that the 1st layer result is still incredibly low, maybe because it is very vital for the 1st layer to focus on passing information so that later layers can learn well. After these two experiments, we want to know what causes the bad performance of bottom layers. Is the simultaneous training of 12 exits too burdening, or the bottom layers themselves cannot master both passing information and classification. So in Fig. 4(c), we place only 6 exits at every other layer. It shows that the 2nd, 4th, and 6th layers perform similarly



(a) (b) (c)

(d) (e) (f)

Fig. 4: (a) – (d) Results on MSCOCO for uni-task CMSS training with multiple ITM exits at different multi-modal encoder layers: (a) two-stage training (training with the last layer ITM head and then linear probing the other 11 heads); (b) training with 12 ITM heads; (c) training with 6 ITM heads at every other layer; (d) training with 6 ITM heads at top 6 layers. (e) Results on MSCOCO of multi-task trained VEMO-sum with ITM exits at top 6 layers. (f) Evaluation time for exiting at each of the top 6 layers of VEMO-sum on MSCOCO 5K test set, with one NVIDIA A100 GPU. Note that the image size is 224 × 224 in (a) – (d), while 384 × 384 in (e) – (f).

to those in Fig. 4(b), indicating their own capability limitations. Therefore, we choose to place 6 exits at top 6 layers of the multi-modal encoder. Results in Fig. 4(d) are more closer to ideal.

In Fig. 4(e) and (f), we present the results and evaluation time on MSCOCO of multi-task trained VEMO-sum, respectively. We can see that all 6 layers of the multi-modal encoder perform better than dual-encoder. As expected, using more layers gets higher results but costs more time. Selectable ITM exits blur the boundary between recall and ranking, making the model more elastic.

4.5 Further Evaluation

To demonstrate the ability of VEMO on different search tasks other than semantic search, we resort to two types of text-to-image search: (1) entity-based image search (EIS) – given a named entity, EIS aims to find images containing this entity; (2) character-based image search (CIS) – given a piece of text, CIS aims to find images that literally have this text in them. For EIS, we randomly sample 5,000 images with their captions from GoodNews [2], which is originally an image-text dataset collected from New York Times. For each image, we employ a strong news NER model [44] to extract named entities from its paired caption, which are regarded as the "ground-truth" entities for this image. We name this processed dataset as GoodNews-5K. For CIS, we directly use an STS dataset CTW1500 [25], which consists of 1,500 images (each image has several curved

Table 3: Results for EIS on GoodNews-5K and CIS on CTW1500.

Method	# PT	Search	GoodNews-5K				CTW1500			
	Images	Type	R@1	R@5	R@10	R@Mean	R@1	R@5	R@10	R@Mean
BLIP-Base	14M	CMSS	1.10	3.42	5.00	3.18	1.19	3.65	5.53	3.46
BLIP-Base	129M	CMSS	1.88	4.51	6.21	4.20	11.92	21.35	26.73	20.00
BLIP-Large	129M	CMSS	2.26	6.05	8.40	5.57	15.24	26.98	32.24	24.82
VEMO-sum	14M	CMSS	1.72	3.56	4.73	3.34	2.61	5.27	6.53	4.80
VEMO-sum	14M	EIS/CIS	**28.67**	**38.15**	**40.48**	**35.77**	**30.24**	**42.46**	**46.33**	**39.68**

text annotations). As a result, GoodNews-5K and CTW1500 are essentially of the same format, *i.e.*, each image has several short text annotations.

We adopt three BLIP models [15] as compared methods for EIS and CIS by directly conducting CMSS. For our VEMO-sum, besides directly conducting CMSS, we can employ the NER/STS ability to address EIS/CIS more gracefully. Specifically, for EIS, VEMO-sum first extracts named entities from all captions as candidates. Then for each entity query, we use the uni-modal text encoder of VEMO-sum to calculate text-to-text similarities between it and all candidates. Finally, we assign the similarity scores to each entity candidate's corresponding image and get the search results. Similarly, for CIS, VEMO-sum first recognize texts in all images as candidates. Then we calculate text-to-text similarities, assign them to corresponding images, and finally obtain the evaluation results.

We report R@1, R@5, R@10, and R@Mean in Table 3, which clearly shows that semantic search is not suitable for either EIS or CIS. On the contrary, VEMO-sum with NER/STS ability outperforms strong semantic search methods (even with large models and 129M training data) by huge margins, validating the general applicability of our VEMO framework on different search tasks.

5 Conclusion

In this work, we investigate how to deal with multiple types of search tasks simultaneously with a single multi-modal model. Specifically, we propose a **V**ersatile **E**lastic **M**ulti-m**O**dal model termed VEMO for search-oriented multi-task learning. VEMO integrates cross-modal semantic search, named entity recognition, and scene text spotting in a unified framework, where the latter two can be further adapted to entity-based and character-based image search tasks, respectively. Furthermore, we place multiple image-text matching exits to offer more choices on the effect-efficiency trade-off for cross-modal semantic search. Extensive experiments validate the effectiveness of our VEMO with significantly fewer network parameters. We believe that search-oriented multi-task learning is meaningful, especially for devices with limited resources.

Acknowledgement

This work was supported by National Natural Science Foundation of China (62376274).

References

1. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: International Conference on Computer Vision (ICCV). pp. 1728–1738 (2021)
2. Biten, A.F., Gómez, L., Rusiñol, M., Karatzas, D.: Good news, everyone! context driven entity-aware captioning for news images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12466–12475 (2019)
3. Chen, S., Zhang, Y., Yang, Q.: Multi-task learning in natural language processing: An overview. arXiv preprint arXiv:2109.09138 (2021)
4. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: UNITER: Universal image-text representation learning. In: European Conference on Computer Vision (ECCV). pp. 104–120 (2020)
5. Ch'ng, C., Chan, C.S., Liu, C.: Total-text: toward orientation robustness in scene text detection. International Journal on Document Analysis and Recognition **23**(1), 31–52 (2020)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT). pp. 4171–4186 (2018)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
8. Feng, W., He, W., Yin, F., Zhang, X.Y., Liu, C.L.: Textdragon: An end-to-end framework for arbitrary shaped text spotting. In: International Conference on Computer Vision (ICCV). pp. 9076–9085 (2019)
9. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9729–9738 (2020)
10. Huang, Z., Zeng, Z., Liu, B., Fu, D., Fu, J.: Pixel-BERT: Aligning image pixels with text by deep multi-modal transformers. arXiv preprint arXiv:2004.00849 (2020)
11. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International Conference on Machine Learning (ICML). pp. 4904–4916 (2021)
12. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S.K., Bagdanov, A.D., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., Shafait, F., Uchida, S., Valveny, E.: ICDAR 2015 competition on robust reading. In: International Conference on Document Analysis and Recognition (ICDAR). pp. 1156–1160 (2015)
13. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7482–7491 (2018)
14. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
15. Li, J., Li, D., Xiong, C., Hoi, S.C.H.: BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning (ICML). pp. 12888–12900 (2022)

16. Li, J., Selvaraju, R.R., Gotmare, A., Joty, S.R., Xiong, C., Hoi, S.C.: Align before fuse: Vision and language representation learning with momentum distillation. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 9694–9705 (2021)
17. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A unified MRC framework for named entity recognition. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 5849–5859 (2020)
18. 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 (ECCV). pp. 121–137 (2020)
19. Liao, L., He, X., Zhao, B., Ngo, C.W., Chua, T.S.: Interpretable multimodal retrieval for fashion products. In: ACM International Conference on Multimedia (ACM-MM). p. 1571–1579 (2018)
20. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: International Conference on Computer Vision (ICCV). pp. 2999–3007 (2017)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755 (2014)
22. Liu, B., Liu, X., Jin, X., Stone, P., Liu, Q.: Conflict-averse gradient descent for multi-task learning. In: Annual Conference on Neural Information Processing Systems (NeurIPS). pp. 18878–18890 (2021)
23. Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1871–1880 (2019)
24. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: ABCNet: real-time scene text spotting with adaptive bezier-curve network. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9806–9815 (2020)
25. Liu, Y., Jin, L., Zhang, S., Luo, C., Zhang, S.: Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recognition **90**, 337–345 (2019)
26. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR) (2019)
27. Lu, H., Fei, N., Huo, Y., Gao, Y., Lu, Z., Wen, J.: COTS: collaborative two-stream vision-language pre-training model for cross-modal retrieval. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15671–15680 (2022)
28. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: European Conference on Computer Vision (ECCV). pp. 67–83 (2018)
29. Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., Fetaya, E.: Multi-task learning as a bargaining game. In: International Conference on Machine Learning (ICML). pp. 16428–16446 (2022)
30. Nayak, P.: MUM: A new AI milestone for understanding information. Google Blog (2021), https://www.blog.google/products/search/introducing-MUM/
31. Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., Khlif, W., Luqman, M.M., Burie, J., Liu, C., Ogier, J.: ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. In: International Conference on Document Analysis and Recognition (ICDAR). pp. 1454–1459 (2017)

32. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: International Conference on Computer Vision (ICCV). pp. 2641–2649 (2015)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML). pp. 8748–8763 (2021)
34. Rafailidis, D., Manolopoulou, S., Daras, P.: A unified framework for multimodal retrieval. Pattern Recognition **46**(12), 3358–3370 (2013)
35. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research **21**, 140:1–140:67 (2020)
36. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I.D., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 658–666 (2019)
37. Sang, E.F.T.K., Meulder, F.D.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Conference on Natural Language Learning (CoNLL). pp. 142–147 (2003)
38. Sun, S., Chen, Y.C., Li, L., Wang, S., Fang, Y., Liu, J.: LightningDOT: Pre-training visual-semantic embeddings for real-time image-text retrieval. In: Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT). pp. 982–997 (2021)
39. Tay, Y., Zhao, Z., Bahri, D., Metzler, D., Juan, D.: HyperGrid Transformers: Towards a single model for multiple tasks. In: International Conference on Learning Representations (ICLR) (2021)
40. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-task learning for dense prediction tasks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3614–3633 (2022)
41. Wang, M., Li, H., Tao, D., Lu, K., Wu, X.: Multimodal graph-based reranking for web image search. IEEE Transactions on Image Processing **21**(11), 4649–4661 (2012)
42. Wang, P., Zhang, C., Qi, F., Liu, S., Zhang, X., Lyu, P., Han, J., Liu, J., Ding, E., Shi, G.: PGNET: Real-time arbitrarily-shaped text spotting with point gathering network. In: AAAI Conference on Artificial Intelligence (AAAI). pp. 2782–2790 (2021)
43. Wang, W., Bao, H., Dong, L., Wei, F.: Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. arXiv preprint arXiv:2111.02358 (2021)
44. Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., Tu, K.: Improving named entity recognition by external context retrieving and cooperative learning. In: Joint Conference of Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP). pp. 1800–1812 (2021)
45. Wen, K., Xia, J., Huang, Y., Li, L., Xu, J., Shao, J.: COOKIE: Contrastive cross-modal knowledge sharing pre-training for vision-language representation. In: International Conference on Computer Vision (ICCV). pp. 2208–2217 (2021)
46. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: Deep contextualized entity representations with entity-aware self-attention. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6442–6454 (2020)

47. Zeng, B.: Go beyond the search box: Introducing multisearch. Google Blog (2022), https://blog.google/products/search/multisearch/
48. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., Gao, J.: VinVL: Revisiting visual representations in vision-language models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5579–5588 (2021)
49. Zhang, X., Su, Y., Tripathi, S., Tu, Z.: Text spotting transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9509–9518 (2022)
50. Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering **34**(12), 5586–5609 (2022)
51. Zhu, E., Li, J.: Boundary smoothing for named entity recognition. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 7096–7108 (2022)
52. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (ICLR) (2021)