Mask4Align: Aligned Entity Prompting with Color Masks for Multi-Entity Localization Problems

Haoquan Zhang[1] Ronggang Huang[1] Yi Xie[1*] Huaidong Zhang[1*]
[1]School of Future Technology, South China University of Technology
haoquanzhang@outlook.com rghuang@gmail.com ftyixie@mail.scut.edu.cn huaidongz@scut.edu.cn

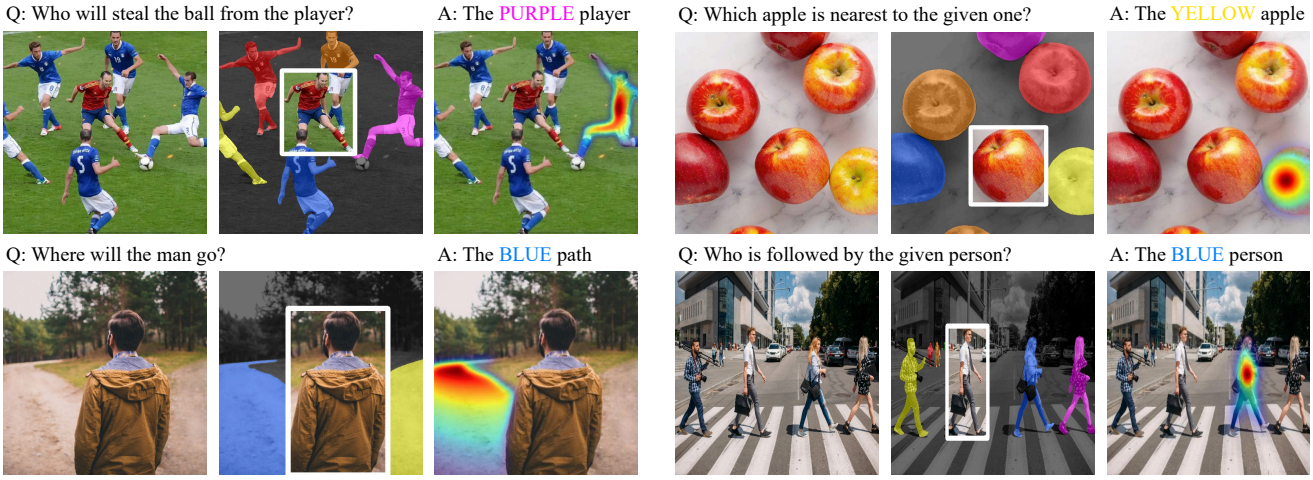

Figure 1. The figure shows six examples of multi-entity localization in Visual Question Answering (VQA). Each scenario includes a question, the original image, an image with color masks, an answer with color, and the corresponding visualized results.

Abstract

In Visual Question Answering (VQA), recognizing and localizing entities pose significant challenges. Pretrained vision-and-language models have addressed this problem by providing a text description as the answer. However, in visual scenes with multiple entities, textual descriptions struggle to distinguish the entities from the same category effectively. Consequently, the VQA dataset is limited by the limitations of text description and cannot adequately cover scenarios involving multiple entities. To address this challenge, we introduce a Mask for Align (Mask4Align) method, which can determine the entity's position in the given image that best matches the user-input question. This method incorporates colored masks into the image, enabling the VQA model to handle discrimination and localization challenges associated with multiple entities. To process an arbitrary number of similar entities, Mask4Align is designed hierarchically to discern subtle differences, achieving precise localization. Since Mask4Align directly utilizes pre-trained models, it does not introduce additional training overhead. Extensive experiments conducted on both the gaze target prediction task dataset and our proposed multi-entity localization dataset showcase the superiority of Mask4Align. Code and dataset are available at https://github.com/HaoquanZhang/mask4align.

1. Introduction

Visual Question Answering (VQA) [11] combines computer vision and natural language processing to answer questions about images, often presented in natural language, covering various aspects of image content. VQA systems fuse image features with textual context to comprehend both the visual content and question semantics, delivering precise answers. In recent years, VQA has gained attention for its practical applications in human-computer interaction [6, 10] and image retrieval [39–43].

While the design of VQA datasets incorporates a comprehensive range of possible scenarios, tailored questions, and accurate answers, its structure still does not fully simulate human VQA behaviors in daily life. For instance, in

*Corresponding authors

scenarios where multiple entities with similar features are present, it becomes challenging to linguistically describe and precisely locate a specific entity. Unlike humans, who can use their hands to 'point out' an object, existing VQA datasets lack a mechanism for explicit localization in such multi-entity scenarios.

Inspired by the human behavior of "pointing out" entities, we propose the Mask for Align (Mask4Align) method. This approach introduces color information to entities in images and generates corresponding candidate answers described with colors. It successfully leverages the VQA modules of pre-trained vision-language models to address multi-entity localization problems, indirectly achieving the operation of "pointing out" objects.

For a multi-entity problem, the Mask4Align method necessitates user input in three parts: the question, the subject's location within the question, and an image depicting the question scene. In the initial stage, working under the assumption that the answer to the input question must belong to one of the entities in the scene, we employ a tagging tool, RAM [51], to extract tags for all entities present. These tags serve as candidate answers for the first input to the VQA module. After retrieving the target answer tag in the first Q&A stage, in the subsequent stage, we input it into a segmentation model [16, 22] to acquire masks corresponding to the entity by utilizing its tag. CPT [45] applies color information to images in various ways, aiding in the creation of corresponding linguistic descriptions for fine-tuning visual language models. Inspired by their work, we utilize overlays of various colors as masks on scene images, obtaining descriptions of color alongside entity names. This process facilitates precise descriptions of entities that may initially be similar and difficult to differentiate through natural language, leveraging the additional semantic information provided by color masks. Consequently, obtaining a well-aligned second batch of candidate answers corresponding to colors becomes straightforward. Finally, we reintroduce the same question, the second batch of candidate answers and the image enhanced with colored masks, into the pre-trained VQA module to obtain the ultimate answer. This yields the entity mask corresponding to the final answer, effectively localizing the target entity.

In this paper, our key contributions are threefold:

- We explored the multi-entity localization problem for the first time in visual question answering and proposed a dedicated multi-entity localization dataset, namely, ME-VQA.
- We propose a method, mask for align (Mask4Align) for the multi-entity localization problem without extra training costs and known candidate answers, which cleverly introduces additional colors into the image, enabling the semantic information they bring to the accurately local entity.
- Extensive experiments demonstrate that our method has achieved usable performance and has been extensively tested in the wild across numerous scenarios.

2. Related Work

2.1. Vision-Language Model Pretraining

Pretrained vision-language models (VLM) play a crucial role in discriminative tasks, such as image-text contrastive (ITC) and image-text matching (ITM), and contribute significantly to the comprehensive understanding of vision and language modalities. The model needs to align and harmonize the representation spaces of both visual and language aspects of the same semantics. For instance, [25], ALBEF [18], and subsequent studies [35, 44] employ cross-modal contrastive learning. This involves projecting both image and language information into a shared and structured semantic space. While excelling in image-text retrieval tasks, many existing models [20, 23, 29, 32, 52] exhibit limitations in capturing interactions between image and text for more complex vision-language tasks such as Visual Question Answering (VQA) [11], Visual Entailment (VE) [38], and Natural Language for Visual Reasoning (NLVR^2) [30]. In this paper, we employed the CLIP and fine-tuned VQA module ALBEF as the VQA modules within Mask4Align.

2.2. Visual Question Answering

In recent years, Visual Question Answering (VQA) has garnered significant attention, drawing researchers from diverse fields to engage in in-depth explorations. Current investigations in the research domain of VQA can be broadly segmented into three main categories: refinement of visual features [3, 13, 28, 50], advancements in model architectures for heightened computational capabilities [12, 15, 19, 47, 48], and the pursuit of more effective learning paradigms [4, 7, 17, 20, 23, 31, 53]. Notably, a prevailing trend among cutting-edge VQA methodologies is the adoption of the Transformer architecture [33]. Through the integration of vision-language pretraining on extensive datasets, these approaches have achieved noteworthy milestones, often rivaling or surpassing human-level performance across various benchmark assessments [2, 34, 46, 49].

3. Method

The proposed Mask4Align framework is in two-stage design, as illustrated in Fig. 3. The VQA model fintuned from pre-trained VLMs is one of the core components of our method. We describe the structure of the VQA module and how we construct the question prototype in Section 3.1. In the first stage of Mask4Align (Section 3.2), we use a tagging tool, Recognize Anything Model (RAM [51]) to obtain the tags of all entities in the scene. Then, we obtain the target entity tag through the first question construction and

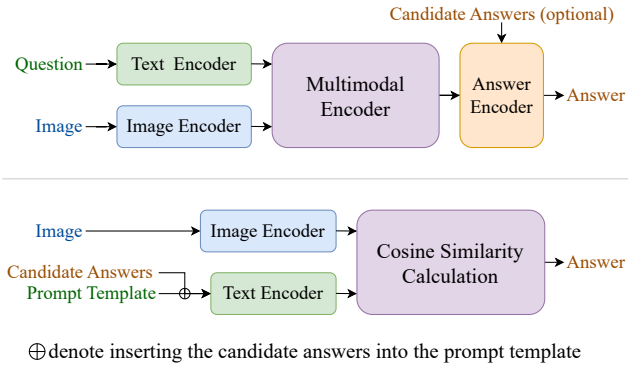

Figure 2. Illustration of the VQA modules. The structure diagram above is the structure of most VQA modules in pre-trained VLMs (such as ALBEF [18], CoCa [46] and BEiT [36]). The structure diagram below is the structure of the CLIP model [25], by using the prompt template obtained from the question, we can also use it as a VQA module.

answering. In the second stage (Section 3.3), we solve the multi-entity localization problem with the novel colorization and answering mechanism, which will be explained in detail in Section 3.3.1. Finally, to handle the dense entity scenes and refine the localization results, we introduce a hierarchical colorization algorithm based on spatial clustering, which aims to progressively localize the target entity in the image (Section 3.3.2).

3.1. The VQA Module

In the following sections (Section 3.2 and Section 3.3), we use the VQA module fine-tuned on the Visual Question Answering [11] dataset as an important component of our method, which is based on pre-trained VLMs.

A common VQA structure is shown in Fig. 5, which can take a question and a corresponding image as inputs and return multiple answer-confidence pairs as outputs (the answer decoder of ALBEF allows candidate answers as input). The CLIP model [25], which is an image-text matching model, can also indirectly accomplish the zero-shot VQA task, but it still requires candidate answers. One possible approach to use CLIP for VQA is to transform a question into a declarative statement by using a prompt template, and then compare the candidate answers with the image input. For example, the question "`Where is the man looking?`" can be converted into the statement "`The man is looking at {Entity}`", where `{Entity}` is a placeholder for the possible answers. The VQA task can be achieved by filling in the candidate answers in the placeholder of the statement and selecting the entity with the highest similarity score with the image input as the answer.

Generally, a VQA module $\mathcal{M}_{vqa}$ require a question q, a batch of candidate answers $\mathcal{A}$ and an image i as inputs, and obtain the final answer a^*:

$$a^* = \mathcal{M}_{vqa}\left(i, q, \mathcal{A}\right). \tag{1}$$

3.2. First Stage: Target Entity Acquiring

3.2.1 Acquiring Tags of Scene Entities

For an entity localization VQA task, the subject of the question must interact with the entities in the image scene, so the first batch of candidate answers $\mathcal{A}_t$ must correspond to the entities in the scene. Therefore, we apply a tagging module RAM [51], denoted as $\mathcal{M}_{tag}$. RAM requires only the scene image i_o as input to perform automatic annotation and obtain scene tags, which are used as candidate answers later:

$$\mathcal{A}_t = \mathcal{M}_{tag}\left(i_o\right). \tag{2}$$

3.2.2 First Question Answering for Target Entity's Tag

After receiving the user's question q and the location bounding box b of the subject, we outline a bounding box around the subject in the image i_o resulting in an image i_b with the added bounding box.

Subsequently, we introduce an adverbial phrase, denoted as t_a, which appends to the original question q_o to modify it. The modified question q_m is then obtained. Alongside $\mathcal{A}_t$, and image with subject's bounding box (i_b), we input them into the $\mathcal{M}_{vqa}$ module. Consequently, we can get the candidate answers sorted by their confidence score. We consider the answer a_t^* with the highest confidence as the one corresponding to the final target entity in the first stage. The above process is formulated by the following equations:

$$\begin{aligned} q_m = q_o + t_a, \quad i_b = i_o \oplus b, \\ a_t^* = \mathcal{M}_{vqa}\left(i_b, q_m, \mathcal{A}_t\right), \end{aligned} \tag{3}$$

where $\oplus$ is denoted as the overlaying process of adding a bounding box onto the image.

3.3. Second Stage: Multi-Entity Localization

With SAM's robust segmentation capabilities, denoted as $\mathcal{M}_{sam}$, we can derive the mask associated with the designated target entity tag a_1^*. In cases where a singular mask s is acquired, it indicates the absence of a multi-entity task within this context, signifying the successful accomplishment of the target entity localization objective. Conversely, if multiple masks are obtained, the scenario is characterized as a multi-entity task. Denote the set of masks as $S = \{s_i\}_{i=1}^{n_s}$, where s_i and n_s refers to the i-th mask and number of masks, respectively:

$$S = \mathcal{M}_{sam}\left(i_o, a_1^*\right). \tag{4}$$

In the subsequent sections, we will clarify our approach for addressing multi-entity tasks.

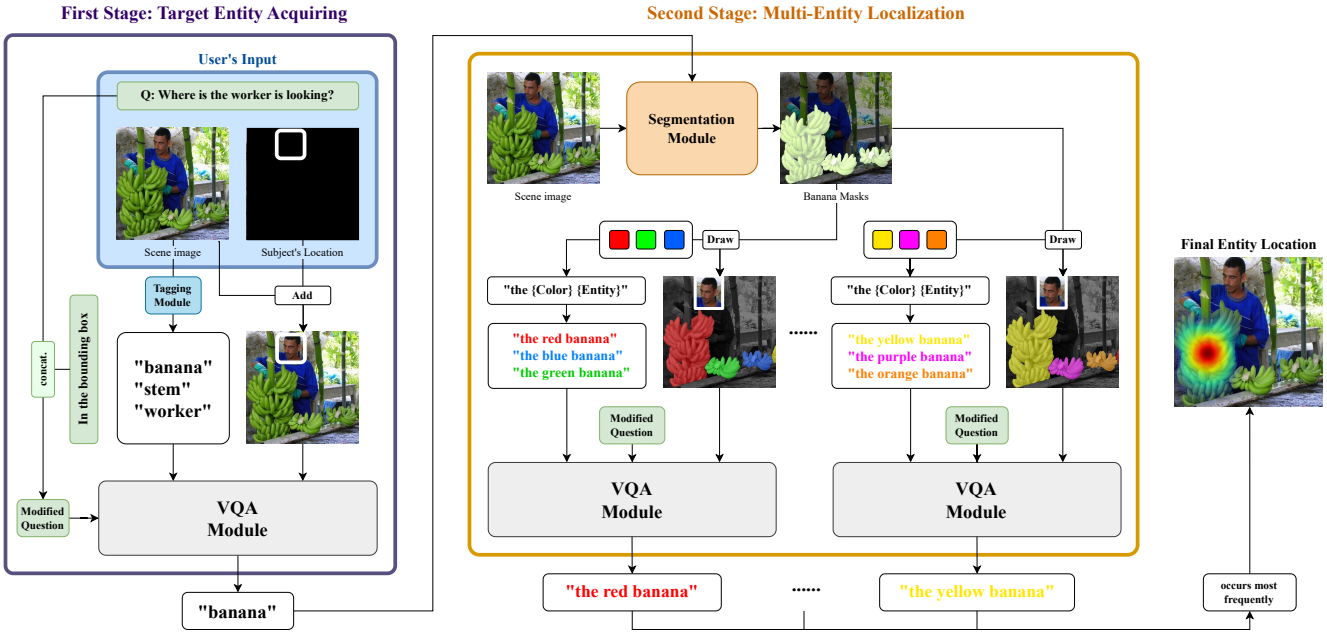

Figure 3. Illustration of our method. Our proposed method consists of two stages. In the first stage, we use the entity tags obtained by a tagging tool as candidate answers and then apply a VQA module to identify the tag corresponding to the target entity. In the second stage, we use a segmentation module to segment the masks that correspond to the target entity tag. Then, we generate images with different colored masks and corresponding batches of candidate answers and feed them along with the question to the VQA module to obtain the answers. Finally, we select the mask that has the highest frequency in the answers as the prediction result for the target entity.

3.3.1 Mask4Align: Color Masking and Aligned Candidate Answers

In most multi-entity situations, natural language struggles to describe entities similar in shape and texture accurately. We propose Mask4Align, an effective multi-entity localization module that leverages the entity discrimination ability of pre-trained VLMs by introducing colored masks and well-aligned prompts. In this section, we will elaborate our operational procedures in detail.

Grayscale Background: To prevent the information conflict between the original colors of entities in the scene image i_o and the colorized masks, we convert the i_o to a grayscale image, denoted as i_g:

$$i_g = f_{rgb2gray}\left(i_o\right), \tag{5}$$

where the $f_{rgb2gray}$ denotes the function converting an RGB image to a grayscale image. To facilitate later overlay with colorful masks, i_g is kept in RGB format.

Color Selection: Next, we will overlay masks onto the i_g using various colors. To ensure pre-trained VLMs comprehend forthcoming color information introduced through masks, our color selection follows these principles:

- Maintain consistency with daily natural language descriptions to prevent model confusion between colors and their corresponding unusual expressions.
- Uncommon colors are avoided, considering the limited exposure of pre-trained VLMs to corresponding texts during training.
- Avoid simultaneous usage of different shades of the same color (e.g., red and dark red) since masks are applied to a grayscale image, and the final color shade may not precisely align with preset color values.

Based on the above considerations, we ultimately selected six colors and corresponding RGB values, as shown in Table 1. We use n_c to denote the number of colors. The set of color expressions is denoted as $E = \{e_j\}_{j=1}^{n_c}$, and the set of color value expressions is denoted as $V = \{(r_j, g_j, b_j)\}_{j=1}^{n_c}$, where the indices are aligned such that e_j corresponds to the RGB values (r_j, g_j, b_j).

Color	RGB Value	Color	RGB Value
Red	(255,0, 0)	Yellow	(255, 255, 0)
Green	(0, 255, 0)	Purple	(255, 0, 255)
Blue	(0, 77, 255)	Orange	(255, 127, 0)

Table 1. The colors and corresponding RGB values.

Answer Template: Subsequently, we use a prompt template a like `"the {Color} {Entity}"` to establish a

correspondence between the entity and its associated color mask. Here, {Color} within the prompt template represents the color expressions e_j, while {Entity} denotes the target entity tag a_1^* derived from the initial question answering as follows:

$$a_i = \text{“the } + e_i \ + a_1^*\text{”}, (e_i \in E), \\ A = \{a_i \mid e_i \in E\}_{i=1}^{n_s}. \tag{6}$$

Generation of Images with Colorful Masks and Candidate Answer Batches: First, we define $\oplus$ as the process of overlaying images of the same size. The elements s_i in the set S are masks representing the original image with the same size, with values of 0 or 1. To assign colors to these masks, we define $\odot$ to represent the multiplication process between s_i and (r_j, g_j, b_j). We use the $\sum$ to express the process of overlaying masks. Subsequently, we multiply i_g by the accumulated colorful masks and their corresponding weights, α and β, where $\alpha + \beta = 1$, resulting in the image with colorful masks. After that, to outline the subject in the image, we overlay the colorful subject' crop i_s on the image and draw the white bounding box b around it. The final modified image is denoted as i_c as follows:

$$i_c = \alpha i_g \oplus \beta \sum_{i,j=1}^{n_s} (s_i \odot (r_j, g_j, b_j)) \oplus i_s \oplus b. \tag{7}$$

The corresponding set of candidate answers A aligned with the order of masks can also be obtained as follows:

$$A_c = \{a_i \mid e_i \in E\}_{i=1}^{n_s}. \tag{8}$$

In summary, the above formulas can be mapped into a function f_{m4a}, which can produce pairs of candidate answer groups and the final modified images. Specifically, The function f_{m4a} that takes l,S, i_o, b, E, V and a_1^* as inputs and yields a pair of i_c and A_c as follows:

$$(i_c^l, \mathcal{A}_c^l) = f_{m4a}(l, S, i_o, b, E, V, a_1^*), \tag{9}$$

where the range of l is from 0 to n_s - 1, and it is an integer.

3.3.2 Second Question Answering for Target Entity's Localization

Since we only used six colors to draw color masks on the image, if the number of masks exceeds the number of the color n_s, the existing colors cannot uniquely represent each mask. Therefore, we designed a method called Recursive-Grouping-VQA (RG-VQA) then shown in Fig. 4 with the multiple entity masks as input, which will be recursively called until it returns the final target entity mask. The logic of the method is as follows:

If the number of entity masks is greater than n_c (shown in the left part of the Fig. 4), we first use the K-means algorithm to divide the masks into six groups, each containing some similar masks. Then we call the function f_{m4a} defined in Eq. (9) (this function has been already overloaded to accept mask group as input), and get an image with group-wise color masks and a list of candidate answers. For each image and answer combination, we use the VQA module mentioned in Section 3.1 to get the most likely answer, which represents a list of mask groups. By changing the order of using colors, we perform the above operations multiple times, and we get a list containing multiple mask groups, and we choose the entity with the highest frequency for subsequent operations. Then recursively apply the algorithm with the final mask group as input.

If the number of masks is less than or equal to n_c (shown in the left part of the Fig. 4), we call the function f to get an image that covers the mask, and a list of possible answers. For each image and answer combination, we use the VQA module to get the most likely answer, which represents the category of the mask. By changing the order of colors, we also perform the above operations multiple times. After getting a list containing multiple masks, we choose the one with the highest frequency as the final target entity mask.

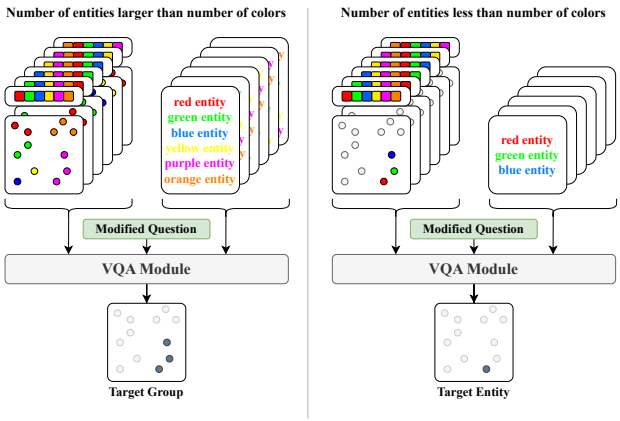

Figure 4. The diagram succinctly illustrates the process of our second question answering iteration. The left side of the image corresponds to the scenario where the number of entities n_s exceeds the number of colors n_c. Initially, we employ k-means clustering to categorize the entities into six classes, followed by a group-wise VQA process. This process yields the entity group with the highest execution performance. On the right side of the image corresponds to the scenario where n_s is less than n_c. In this case, an entity-wise VQA process is conducted, ultimately resulting in the identification of the target entity.

4. Experiments on Gaze Target Prediction

In this Section, we evaluate our method on the GazeFollow [26] dataset and the proposed ME-VQA dataset, for the task of gaze target prediction and present the experimental results. We first provide a brief introduction to the datasets and performance metrics. Then, we conduct ablation and

comparison experiments to prove the effectiveness of our method. Finally, we analyze the sensitivity of the hyper-parameter α and some failure cases.

4.1. Datasets and Evaluation Metrics

GazeFollow. In order to understand human actions by figuring out their gaze target location, Recasens et al [27]. presented the first dataset called GazeFollow, which includes third-person view 2D images with gaze target annotations. dataset contains 130,339 individuals across 122,143 images, each annotated with gaze locations. Images are sourced from popular datasets, such as MS COCO [21] and PASCAL [8]. The GazeFollow test set includes 4,782 individuals, each annotated by 10 annotators.

ME-VQA. This dataset is proposed for the multi-entity localization problem, comprising 5 categories, each containing 15 images, for a total of 75 images. Moreover, each image corresponds to two questions, and each question contains a human-annotated mask to evaluate pixel-level errors, as shown in Fig. 5. We also provide two examples for each of the five categories as depicted in Fig. 7.

Figure 5. The partial presentation for our ME-VQA dataset.

Evaluation Metrics. Following previous gaze target estimation works [5, 26], we employ evaluation metrics including AUC, L2 Distance, and Angular error, proposed by Judd et al. [14], to assess predictive accuracy and model capability. In contrast, following previous gaze target estimation works [37], ME-VQA evaluation metrics comprise S-measure (S_α) [9], maximum F-measure (F_{max}) [1], and Mean Absolute Error (MAE) [24].

4.2. Ablation Experiments

As shown in Table 2 and Table 3, we conduct ablation experiments across two VQA modules (i.e., CLIP [25] and ALBEF [18]) that allow inputting candidate answers on GazeFollow [26] and ME-VQA datasets. "CLIP" means that CLIP is used as the VQA module to find the answer entity. "CLIP + Random" represents that an answering entity is randomly selected when CLIP is employed as the VQA module. "CLIP + Mask4Align w/o RG-VQA" indicates that the RG-VQA is removed from Mask4Align when CLIP is employed as the VQA module. Similarly, "ALBEF", "ALBEF + Random", and "ALBEF + Mask4Align w/o RG-VQA" denote representations when ALBEF is employed as the VQA module.

In Table 2, on GazeFollow [26] test set, "CLIP + Mask4Align w/o RG-VQA" surpasses "CLIP" by 0.027 AUC, 0.024 Dist, and 2.30 Ang. Similarly, "ALBEF + Mask4Align w/o RG-VQA" outperforms "ALBEF" by 0.054 AUC, 0.034 Dist, and 6.5 Ang. These results show that integrating additional colors into images enhances the VQA module's capability to localize entities. Moreover, the performance of these VQA modules is further enhanced by using RG-VQA. For example, "ALBEF + Mask4Align" outperforms "ALBEF + Mask4Align w/o RG-VQA" by 0.002 AUC, 0.004 Dist, and 0.5 Ang.

As our method is specifically designed for multi-entity localization, its effectiveness may not be adequately assessed on standard VQA datasets. Therefore, we conduct ablation experiments on a dedicated multi-entity dataset, as shown in Table 3. From Table 3, we observe that our method achieves more significant results on the multi-entity dataset. Specifically, "CLIP + Mask4Align" exceeds "CLIP" by 0.248 S_α, 0.009 F_{max}, and 0.309 MAE. "ALBEF + Mask4Align" outperforms "ALBEF" by 0.307 S_α, 0.217 F_{max}, and 0.271 MAE. These above comparison results demonstrate that Mask4Align can effectively improve the VQA module's capability to localize entities in the multi-entity scene.

METHOD	AUC ↑	Dist ↓	MDist ↓	Ang ↓	MAng ↓
CLIP [25]	0.648	0.340	0.257	42.3	30.1
CLIP + Mask4Align w/o RG-VQA	0.675	0.316	0.238	40.0	28.1
CLIP + Mask4Align	**0.677**	**0.312**	**0.234**	**39.5**	**27.8**
ALBEF [18]	0.688	0.300	0.227	39.2	28.2
ALBEF + Mask4Align w/o RG-VQA	0.742	0.266	0.193	32.7	24.1
ALBEF + Mask4Align	**0.745**	**0.263**	**0.191**	**32.4**	**23.8**

Table 2. Ablation study across various VQA models (i.e., CLIP [25] and ALBEF [18]) on the GazeFollow test set [26].

METHOD	S_α ↑	F_{max} ↑	MAE ↓
CLIP + Random	0.408	0.596	0.477
CLIP + Mask4Align w/o RG-VQA	0.625	0.598	0.197
CLIP + Mask4Align	**0.656**	**0.615**	**0.168**
ALBEF + Random	0.514	0.572	0.342
ALBEF + Mask4Align w/o RG-VQA	0.781	0.732	0.088
ALBEF + Mask4Align	**0.821**	**0.789**	**0.071**

Table 3. Ablation study across various VQA models (i.e., CLIP [25] and ALBEF [18]) on ME-VQA.

4.3. Comparison Experiments

To comprehensively evaluate Mask4Align, we explore different variants with various image modifications and corresponding prompt settings on GazeFollow [26]. Specifically, we construct two different variant methods,

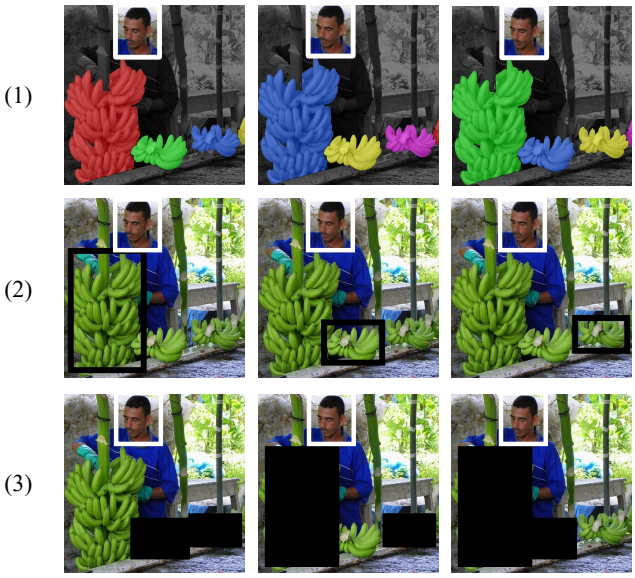

Figure 6. Different variants align methods. (1) Mask4Align, (2) Box4Align, (3) Covering4Align.

namely, Box4Align and Covering4Align, as shown in Fig. Fig. 6. Box4Align selects the target entity by adding a black bounding box around one of the entities. Then we set the prompt input as "`In the white bounding box, the worker is watching at {Entity} in the black bounding box`", and choose the entity with the highest confidence. Covering4Align selects the target entity by covering the other entities with filled rectangles until only one is left. Then we set the prompt input as "`In the white bounding box, the worker is watching at {Entity}`", and choose the entity with the highest confidence.

In Table 4, it is evident that Mask4Align outperforms the other two methods. For example, Mask4Align exceeds Box4Align by 0.020 AUC, 0.021 Dist, and 1.6 Ang. This is attributed to the significant impact that either highlighting a section of the image using a bounding box or erasing certain information by covering a rectangle has on the image's coherence and interpretation. In contrast, Mask4Align adeptly preserves the image's continuity while introducing additional color information without substantially altering its semantics. This preservation allows the VQA module to more effectively interpret the semantic content within the image, thus contributing to Mask4Align achieving superior performance.

4.4. Sensitivity Analysis

In this section, we analyze the sensitivity of the hyperparameters α and β, which represent the weights of the grayscale image i_g background and colorful masks, respectively, when they are overlaid in Eq. (7). Since α and β sum to 1, essentially, this process involves only one hyperparameter, α, where β is defined as $1-\alpha$. We employed two VQA modules, ALBEF [18] and CLIP [25], and conducted evaluations within the range of α from 0.3 to 0.7, with a step size of 0.1. The performances are shown in Table 5.

METHOD	AUC ↑	Dist ↓	MDist ↓	Ang ↓	MAng ↓
CLIP + Mask4Align	**0.675**	**0.316**	**0.238**	**40.0**	**28.1**
CLIP + Box4Align	0.655	0.337	0.247	41.6	29.7
CLIP + Covering4Align	0.653	0.333	0.246	41.9	29.8

Table 4. Performance comparison across different variants method on the GazeFollow test set [26].

From Table 5, we observe that variations in the α value do not notably affect the models' performance. A balanced performance is consistently achieved when α is set to 0.5. Therefore, we adopt 0.5 as the default value for α.

METHOD	α	AUC ↑	Dist ↓	MDist ↓	Ang ↓	MAng ↓
CLIP + Mask4Align	0.3	0.674	0.320	0.240	40.2	28.2
	0.4	0.674	0.318	**0.239**	40.2	28.3
	0.5	**0.675**	**0.316**	0.240	**40.0**	**28.1**
	0.6	0.676	0.318	0.240	40.1	28.2
	0.7	0.673	0.319	0.243	40.3	28.2
ALBEF + Mask4Align	0.3	0.742	0.263	**0.193**	33.0	24.3
	0.4	0.742	0.266	0.198	32.9	24.5
	0.5	**0.742**	**0.266**	**0.193**	**32.7**	**24.1**
	0.6	0.741	0.267	0.197	33.5	24.7
	0.7	0.740	0.265	0.197	33.5	24.7

Table 5. The influence performances of the α value on the GazeFollow test set [26].

4.5. Failure Cases Analysis

In certain cases, the VQA model tends to prefer legal descriptions in the wild, such as choosing "red fruit" instead of the correct answer "orange fruit". The example is shown in Fig. 7. F. We suspect that the reason for this bias might lie in the abundance of red fruit-shaped samples in the training data, leading the model to favor red fruit as the correct choice. Aggregating answers across various queries under different color conditions, as demonstrated in the supplementary material, partially but not completely mitigates this issue.

5. Conclusion

In the context of the Visual Question Answering (VQA) task, the challenge of multi-entity localization is constrained by the limited ability of natural language to describe similar objects, a facet largely overlooked in most research endeavors. Even native VQA modules derived from

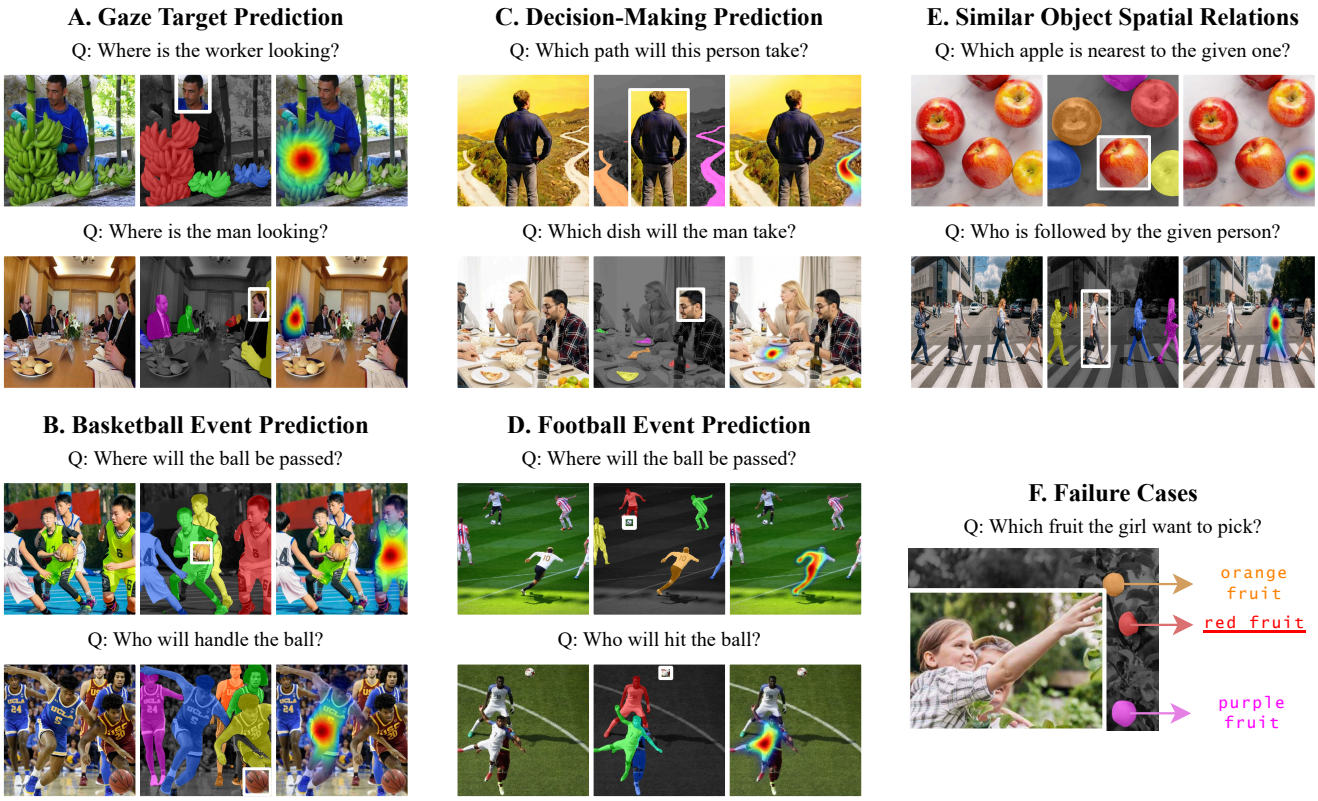

Figure 7. Visualization of five applications on various multi-entity scenarios and one failure example.

pretrained Vision-Language models prove inadequate in addressing this particular issue. Therefore, we introduce the Mask4Align method, which incorporates color information into images and generates candidate answers adorned with corresponding colors. This approach successfully leverages the discerning capabilities of the overlooked VQA modules for multi-entity resolution. Our method yields reliable entity localization results in the gaze target prediction task and demonstrates favorable application outcomes across various scenarios.

Limitations. The method faces the issues as follows:

- The performance of the VQA model directly impacts the final performance.
- Not all models allow direct textual input for candidate answers, and currently, the best-performing VLM in the VQA task does not allow input of candidate answers either.
- The selection of colors is predefined and cannot automatically adapt to different models' preferences.
- There is a lack of extensive datasets specifically tailored for multi-entity tasks, hindering comprehensive training and testing.

Based on these issues, several potential avenues for further investigation emerge:

- To leverage more robust models for addressing multi-entity localization problems, we may need to devise a method to embed candidate answers, potentially incurring additional training costs.
- A question-answering approach could be designed to enable models to select appropriate colors.
- The collection of a larger dataset would facilitate more comprehensive training and testing.

Broader Impact: Our approach demonstrates versatility, applicable to a range of challenges in multi-entity localization within the field of computer vision. As an unsupervised method that doesn't require additional learning, Mask4Align can be directly applied to specific tasks without training. The concept of integrating information akin to Mask4Align holds the potential to stimulate further research within the community, particularly in endeavors concentrating on aligning vision and language.

Acknowledgement. The work is supported by the National Natural Science Foundation of China (No.62302170).

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1597–1604. IEEE, 2009. 6

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[3] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018. 2

[4] 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 *European Conference on Computer Vision*, pages 104–120, 2020. 2

[5] Eunji Chong, Yongxin Wang, Nataniel Ruiz, and James M Rehg. Detecting attended visual targets in video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5396–5406, 2020. 6

[6] M Kalpana Chowdary, Tu N Nguyen, and D Jude Hemanth. Deep learning-based facial emotion recognition for human–computer interaction applications. *Neural Computing and Applications*, 35(32):23311–23328, 2023. 1

[7] Yuhao Cui, Zhou Yu, Chunqi Wang, Zhongzhou Zhao, Ji Zhang, Meng Wang, and Jun Yu. Rosita: Enhancing vision-and-language semantic alignments via cross-and intra-modal knowledge integration. In *ACM International Conference on Multimedia*, pages 797–806, 2021. 2

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2010. 6

[9] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *IEEE/CVF International Conference on Computer Vision*, pages 4548–4557, 2017. 6

[10] Hristijan Gjoreski, Ifigeneia Mavridou, James Archer William Archer, Andrew Cleal, Simon Stankoski, Ivana Kiprijanovska, Mohsen Fatoorechi, Piotr Walas, John Broulidakis, Martin Gjoreski, et al. Ocosense glasses–monitoring facial gestures and expressions for augmented human-computer interaction: Ocosense glasses for monitoring facial gestures and expressions. In *CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2023. 1

[11] 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. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 1, 2, 3

[12] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *IEEE/CVF International Conference on Computer Vision*, pages 804–813, 2017. 2

[13] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276, 2020. 2

[14] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *IEEE/CVF International Conference on Computer Vision*, pages 2106–2113, 2009. 6

[15] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[17] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2

[18] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34:9694–9705, 2021. 2, 3, 6, 7

[19] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *IEEE/CVF International Conference on Computer Vision*, pages 10313–10322, 2019. 2

[20] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137, 2020. 2

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. 6

[22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2

[23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[24] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 733–740, 2012. 6

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3, 6, 7

[26] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? *Advances in Neural Information Processing Systems*, 28, 2015. 5, 6, 7

[27] Adria Recasens, Aditya Khosla, Carl Vondrick, and Antonio Torralba. Where are they looking? *Advances in Neural Information Processing Systems*, 28, 2015. 6

[28] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2022. 2

[29] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, June 2022. 2

[30] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Association for Computational Linguistics*, pages 6418–6428, 2019. 2

[31] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, pages 5100–5111, 2019. 2

[32] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, June 2022. 2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[34] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340, 2022. 2

[35] Teng Wang, Wenhao Jiang, Zhichao Lu, Feng Zheng, Ran Cheng, Chengguo Yin, and Ping Luo. Vlmixer: Unpaired vision-language pre-training via cross-modal cutmix. In *International Conference on Machine Learning*, pages 22680–22690, 2022. 2

[36] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023. 3

[37] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3239–3259, 2021. 6

[38] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019. 2

[39] Yi Xie, Fei Shen, Jianqing Zhu, and Huanqiang Zeng. Viewpoint robust knowledge distillation for accelerating vehicle re-identification. *EURASIP Journal on Advances in Signal Processing*, 2021(1):1–13, 2021. 1

[40] Yi Xie, Hanxiao Wu, Fei Shen, Jianqing Zhu, and Huanqiang Zeng. Object re-identification using teacher-like and light students. In *British Machine Vision Conference*, 2021. 1

[41] Yi Xie, Hanxiao Wu, Jianqing Zhu, and Huanqiang Zeng. Distillation embedded absorbable pruning for fast object re-identification. *Pattern Recognition*, 152:110437, 2024. 1

[42] Yi Xie, Huaidong Zhang, Xuemiao Xu, Jianqing Zhu, and Shengfeng He. Towards a smaller student: Capacity dynamic distillation for efficient image retrieval. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16006–16015, 2023. 1

[43] Yi Xie, Jianqing Zhu, Huanqiang Zeng, Canhui Cai, and Lixin Zheng. Learning matching behavior differences for compressing vehicle re-identification models. In *IEEE International Conference on Visual Communications and Image Processing*, pages 523–526, 2020. 1

[44] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022. 2

[45] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *AI Open*, 5:30–38, 2024. 2

[46] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2, 3

[47] Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. Deep multimodal neural architecture search. In *ACM International Conference on Multimedia*, pages 3743–3752, 2020. 2

[48] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *IEEE/CVF International Conference on Computer Vision*, pages 1821–1830, 2017. 2

[49] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[50] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. 2

[51] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023. 2, 3

[52] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, June 2022. 2

[53] Mingyang Zhou, Licheng Yu, Amanpreet Singh, Mengjiao Wang, Zhou Yu, and Ning Zhang. Unsupervised vision-and-language pre-training via retrieval-based multi-granular alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16485–16494, 2022. 2