Semi-supervised Mixture Model for Visual Language Multitask

Anonymous ICCV submission

Paper ID 17

Abstract

This paper proposes a semi-supervised mixture model Detection-Language-SAM (a Visual Question Answering Model) that detects and segments input images and answers input questions. The model comprises three main components: the classic object detection model You Only Look Once (YOLOv3), the lightweight pre-trained language model DistilBERT, and Segmentation Anything Model (SAM). YOLOv3 is trained using the same training dataset as SAM to obtain a pre-trained object detection model. Secondly, the proposed questions are answered using the pre-trained DistilBERT model. The generated text prompts are input into SAM to segment the final target objects. The proposed model aims to combine large-scale pre-trained visual models with lightweight language models to create a more powerful visual language model that can better accomplish visual language tasks.

1. Introduction

Recently, the combination of large-scale language models and visual models has led to the development of powerful vision-and-language models, which can jointly process visual and textual information. Prompt-based reasoning is an emerging approach that utilizes large-scale visual and language models for zero-shot generalization. This method can help the model adapt and reason quickly when facing new tasks, scenes, or domains. However, fully leveraging text cues to harness the performance of pre-trained visual and language models is a challenging task [1].

To better accomplish complex visual language tasks and leverage the capabilities of large-scale pre-trained models, a semi-supervised mixture model that can perform detection and segmentation on input images is proposed, as well as answer questions. The proposed model consists of the detection model YOLOv3 [2], the language model DistilBERT [3], and the segmentation model SAM [4].

2. Related work

Recently, Visual Question Answering (VQA) has gained attention for its potential applications in image retrieval, captioning, virtual assistants, and medical diagnosis [1]. These models typically employ CNNs for image processing, RNNs or Transformer-based models for question processing, and attention mechanisms to fuse visual and textual information. They achieve state-of-the-art performance on VQA benchmark tests, demonstrating their effectiveness in answering image-related questions [5]. However, these VQA models require supervised training on specific datasets and do not incorporate large-scale pre-trained models.

With the rise of self-supervised learning, more pre-trained visual models based on autoencoders and contrastive learning have emerged. Transformer-based models, initially renowned for their success in natural language processing, have garnered considerable attention in computer vision as well. This is attributed to their capability to establish profound interactions between images and texts, which in turn facilitates robust visual feature representation learning. These models excel in tasks such as image classification, object detection, and image generation [1, 5]. However, these pre-trained large-scale visual models have not yet been combined with Visual Question Answering (VQA) to jointly perform tasks such as detection, segmentation, classification, and question answering [6].

3. Method

The proposed semi-supervised mixture visual language model is shown in Figure 1. This model comprises YOLOv3 [2] for object detection, DistilBERT [3] for language question answering, and SAM [4] for image segmentation. It is capable of performing detection, segmentation, and question-answering tasks. By inputting an image and a question, the model uses visual detection and segmentation models for image detection and segmentation, and the language model for answering questions. The models engage in a collaborative process through text prompts for visual

language tasks. The process is illustrated as follows:

$$(Det, Seg, Det\ \&\ Seg, A) = VLM_{DSA}(In_{img}, Q) \quad (1)$$

where VLM_{DSA} is the proposed model; In_{img} is input image; Q is input question; and $(Det, Seg, Det\ \&\ Seg, A)$ is the output of the detected image, the segmented image, the image after detection and segmentation, and the answer to the question.

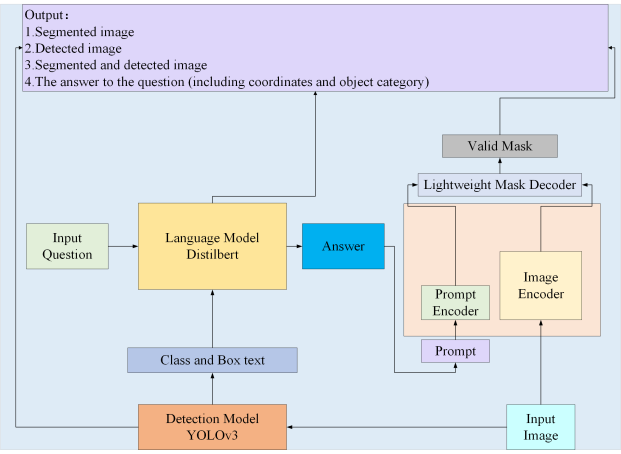

Figure 1. Structure of proposed semi-supervised mixture visual language model (Detection-Language-SAM).

3.1. Visual object detection model

The visual model adopts the classical object detection algorithm YOLOv3, which is mainly used to obtain the category and bounding box information of the objects in the input image. Its model structure mainly includes the backbone network, neck, and head.

The backbone network of YOLOv3 utilizes the Darknet-53, which consists of 53 convolutional layers. Among these layers, the first 52 layers employ a 3 × 3 convolutional kernel with a stride of 1, and Leaky ReLU is used as the activation function. The last layer employs a 1 × 1 convolutional kernel with a stride of 1, and also uses Leaky ReLU as the activation function. Assuming the input feature map is denoted as X, the convolutional operation for the i-th layer can be represented as:

$$X_i = LeakyReLU(Conv(X_{i-1}, w_i + b_i)) \quad (2)$$

where X_{i-1} is the feature map from the previous layer; w_i is the convolutional kernel parameters of the i-th layer; b_i represents the bias term parameters of the i-th layer; $Conv$ is the convolutional operation; and $LeakyReLU$ is the activation function.

The neck of YOLOv3 is primarily used for feature fusion and downsampling operations on the feature map from the backbone network. The neck consists of two convolutional layers, one with a 1 × 1 convolutional kernel, stride of 1, and Leaky ReLU activation function; and the other one with a 3 × 3 convolutional kernel, stride of 2, and Leaky ReLU activation function. Assuming the input feature map is X, the computation of the neck can be represented as:

$$X_{neck} = LeakyReLU(Conv_1(X)) \quad (3)$$

$$X_{neck} = LeakyReLU(Conv_3(X_{neck})) \quad (4)$$

where $Conv_1$ represents a convolution operation with a 1 × 1 kernel; and $Conv_3$ represents a convolution operation with a 3 × 3 kernel.

The head of YOLOv3 is responsible for predicting the bounding box positions and object classes for object detection. The head consists of three feature layers of different scales, which are used for detecting small, medium, and large-sized objects. Each feature layer uses three different-sized convolutional kernels (1 × 1, 3 × 3, and 1 × 1) for predicting the bounding boxes and classes of objects. Additionally, YOLOv3's head utilizes anchor boxes to provide prior information about the size and aspect ratio of the target boxes, which helps improve the accuracy of object detection. Assuming the input feature map is denoted as X, the computation of the head can be represented as:

$$X_{head} = LeakyReLU(Conv_1(X_{neck})) \quad (5)$$

$$X_{head} = LeakyReLU(Conv_3(X_{head})) \quad (6)$$

$$bbox_{pred} = Conv_{1,bbox}(X_{head}) \quad (7)$$

$$class_{pred} = Conv_{1,class}(X_{head}) \quad (8)$$

where $Conv_{1,bbox}$ is the 1 × 1 convolution operation for predicting the bounding box position information; and $Conv_{1,class}$ is the 1 × 1 convolution operation for predicting the class information of the target.

3.2. Language model

In the proposed vision-language model, the language model of choice is DistilBERT. DistilBERT is a lightweight pre-trained language model that is a simplified version of BERT (Bidirectional Encoder Representations from Transformers) model. DistilBERT is obtained by compressing the pre-trained BERT model using distillation technique, to reduce the model size and computational resource requirements while maintaining high performance. Assuming that the input to DistilBERT is a text sequence X, and the output after encoding is denoted as H, the calculation formulas during the fine-tuning stage are as follows:

$$H = P(TE([X + PE])) \quad (9)$$

where $P(\cdot)$ is the calculation of pooling; $TE(\cdot)$ is the calculation of the transformer encoder; and PE represents position encodings.

The calculation of the transformer encoder mainly includes self-attention, feedforward neural network, and layer normalization. Their calculations are as follows.

Self-Attention:

$$Q = XW_Q, K = XW_K, V = XW_V \tag{10}$$

where W_Q, W_K, W_V are learned weight matrices that map the input X to query, key, and value vectors, respectively.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{11}$$

where d_k is the dimensionality of the query and key vectors.

Feedforward neural network:

$$FFN(H) = ReLU(HW_1 + b_1)W_2 + b_2 \tag{12}$$

where W_1,W_2,b_1, and b_2 are the learned weight matrices and bias vectors, respectively.

Layer normalization:

$$y = \gamma * \frac{x - u}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{13}$$

where γ and β are learned scaling and shifting factors, respectively; μ and σ^2 are the mean and variance of H, respectively; and ε is a small constant used to avoid division by zero.

Task-specific fully connected layer:

$$Y = Softmax(H_{task}W_{task} + b_{task}) \tag{14}$$

where H_{task} is the portion of the encoder's output H that is relevant to the task; W_{task} and b_{task} are the weight matrix and bias vector of the task-specific fully connected layer, respectively.

Loss function:

$$L_{task} = -\sum_{N}^{i=1}\sum_{C}^{j=1} Y_{task}(i,j)logY(i,j) \tag{15}$$

where N is the number of samples in the annotated data; C is the number of classes in the task; $Y_{task}(i,j)$ is the true value of the j-th class label for the i-th sample; and $Y(i,j)$ is the predicted output value of the model during the fine-tuning phase.

3.3. Segmentation anything model

In the proposed model, the self-supervised model SAM is selected for high-quality segmentation. SAM consists of three components: an image encoder, a flexible prompt encoder, and a fast mask decoder.

In SAM, a pre-trained Vision Transformer (ViT) is used as the image encoder. ViT includes two main components: Patch Embedding and Transformer Encoder. In the Patch Embedding, the input image I is mapped to Patch Embedding X through linear projection. The calculation of Patch Embedding is shown as follows:

$$X = Patch\_Embedding(I) \tag{16}$$

The Transformer Encoder mainly consists of two parts: self-attention and a feed-forward neural network. Self-attention obtains query vector Q, key vector K, and value vector V from Patch Embedding X through linear projection, as calculated in Equations 17, 18, and 19, respectively. Then, dot product attention is used to compute self-attention scores, representing the relevance of each patch with other patches in the image. Afterward, a feed-forward neural network is independently applied to each Patch Embedding, introducing non-linearity. The calculations of them are shown as follows:

$$Q = Linear(X) \tag{17}$$

$$K = Linear(X) \tag{18}$$

$$V = Linear(X) \tag{19}$$

The Prompt Encoder part includes two types of prompt encoding methods: sparse (points, boxes, texts) and dense (masks). Sparse prompt encoding for points and boxes involves adding positional encodings to learned embeddings for each prompt type. Dense prompt encoding involves convolutional operations to embed the prompts, followed by element-wise addition with the image embeddings. The calculations of them are illustrated as follows:

$$P\_E(p/b) = Pos\_E(p/b) + L\_E(p/b) \tag{20}$$

$$P\_E(d_{masks}) = Conv(d_{masks})I\_E \tag{21}$$

where P_E is prompt embedding; Pos_E is position embedding; L_E is learned embedding; p is points; b is boxes; and d_{masks} is dense masks.

The mask decoder utilizes Transformer decoder blocks and updates the embedding representation using prompt self-attention and cross-attention. Afterward, the image embedding is upsampled, and an MLP maps the output token to a linear classifier, which calculates the foreground probability of the mask at each image location.

4. Experiments and results

4.1. Model training procedure

The training process for the proposed model is divided into three stages. Firstly, the SA-1B dataset is used to train YOLOv3. Then, YOLOv3 and a pre-trained DistilBERT were jointly trained. Finally, YOLOv3, pre-trained DistilBERT, and pre-trained SAM were jointly trained end-to-end.

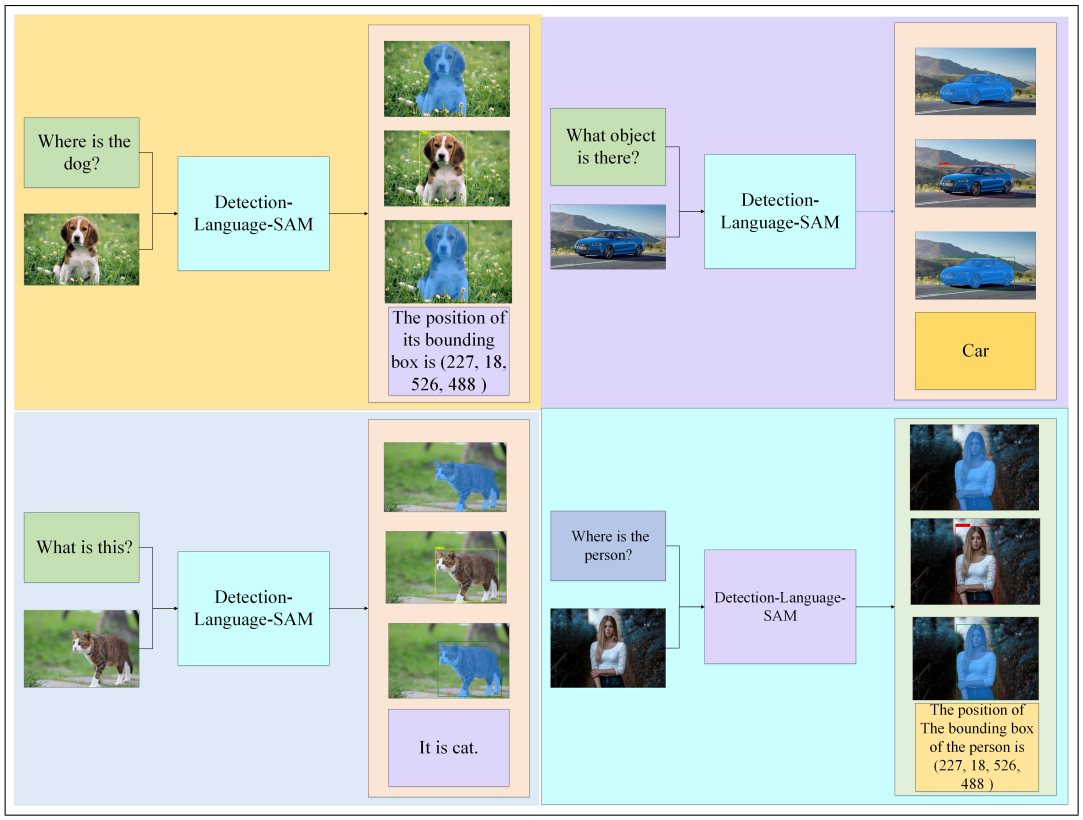

Figure 2. Experiment results of the proposed semi-supervised mixture visual language model (Detection-Language-SAM).

4.2. Evaluation and results

We have completed qualitative experiments, and the results are shown in Figure 2. In Figure 2, by inputting the image and question into the proposed semi-supervised mixture visual language model, we obtained four outputs: the detected image, segmented image, detected and segmented image, and the answer to the input question. Furthermore, we are currently conducting quantitative experiments to evaluate the proposed model from multiple metrics and perspectives.

5. Discussion

We are gonging to make two improvements to enhance the model's performance. Firstly, we aim to enable the model to perform more fine-grained object classification. Secondly, we plan to incorporate the Diffusion model into SAM to improve the segmentation results.

References

[1] Chappuis C, Zermatten V, Lobry S, et al. Prompt-RSVQA: Prompting visual context to a language model for remote sensing visual question answering[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 1372-1381. 1

[2] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018. 1

[3] Sanh V, Debut L, Chaumond J, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter[J]. arXiv preprint arXiv:1910.01108, 2019. 1

[4] Kirillov A, Mintun E, Ravi N, et al. Segment anything[J]. arXiv preprint arXiv:2304.02643, 2023. 1

[5] Mahabadi R K, Zettlemoyer L, Henderson J, et al. Perfect: Prompt-free and efficient few-shot learning with language models[J]. arXiv preprint arXiv:2204.01172, 2022. 1

[6] Zhang Z, Wu W, Sun W, et al. MD-VQA: Multi-dimensional quality assessment for UGC live videos[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 1746-1755. 1