electronics

Article

Multi-Modal Alignment of Visual Question Answering Based on Multi-Hop Attention Mechanism

Qihao Xia [1], Chao Yu [1,2,3,*], Yinong Hou [1], Pingping Peng [1], Zhengqi Zheng [1,2] and Wen Chen [1,2,3]

[1] Engineering Center of SHMEC for Space Information and GNSS, East China Normal University, Shanghai 200241, China; 51205904027@stu.ecnu.edu.cn (Q.X.); 51215904099@stu.ecnu.edu.cn (Y.H.); 51205904079@stu.ecnu.edu.cn (P.P.); zqzheng@ee.ecnu.edu.cn (Z.Z.); wchen@sist.ecnu.edu.cn (W.C.)
[2] Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China
[3] Key Laboratory of Geographic Information Science, Ministry of Education, East China Normal University, Shanghai 200241, China
* Correspondence: cyu@sist.ecnu.edu.cn; Tel./Fax: +86-215-434-5028

Abstract: The alignment of information between the image and the question is of great significance in the visual question answering (VQA) task. Self-attention is commonly used to generate attention weights between image and question. These attention weights can align two modalities. Through the attention weight, the model can select the relevant area of the image to align with the question. However, when using the self-attention mechanism, the attention weight between two objects is only determined by the representation of these two objects. It ignores the influence of other objects around these two objects. This contribution proposes a novel multi-hop attention alignment method that enriches surrounding information when using self-attention to align two modalities. Simultaneously, in order to utilize position information in alignment, we also propose a position embedding mechanism. The position embedding mechanism extracts the position information of each object and implements the position embedding mechanism to align the question word with the correct position in the image. According to the experiment on the VQA2.0 dataset, our model achieves validation accuracy of 65.77%, outperforming several state-of-the-art methods. The experimental result shows that our proposed methods have better performance and effectiveness.

Keywords: multi-modal alignment; multi-hop attention; visual question answering; feature fusion

Citation: Xia, Q.; Yu, C.; Hou, Y.; Peng, P.; Zheng, Z.; Chen, W. Multi-Modal Alignment of Visual Question Answering Based on Multi-Hop Attention Mechanism. *Electronics* **2022**, *11*, 1778. https://doi.org/10.3390/electronics11111778

Academic Editors: Leonardo Galteri, Claudio Ferrari and Stefanos Kollias

Received: 28 April 2022
Accepted: 30 May 2022
Published: 3 June 2022

1. Introduction

With the development of computer vision and natural language processing, deep learning models can deal with tasks that require understanding and reasoning about information from images and texts, e.g., image captioning [1], visual question answering (VQA) [2], and visual dialog [3,4]. Image captioning is a task that needs models to describe a given image and output the image's description with text. VQA is a multi-modal task that requires a complex reasoning process. Given an image and a question related to the image, the model needs to reason about the image–question pair and output the answer closest to the real answer. As an important part of artificial intelligence, more and more researchers have been paying attention to the VQA task in recent years.

More and more fields are considering the application of VQA at the moment. Intelligent medical treatment is regarded as a field with great application potential for VQA. The medical image understanding of AI and medical image-related question answering can help doctors to make diagnoses. The related med-VQA datasets, such as VQA-RAD [5] and PathVQA [6], have been proposed recently, improving the specificity of this challenge. The authors of [7] propose a med-VQA deep learning approach exploiting a multi-modal question-centric strategy. In [8], the authors turn the complex med-VQA problem into multiple simpler problems through classification and a generative model. Assistance of

blind and visually impaired individuals is also one of the objectives of several VQA applications. The related dataset [9] consists of 31,000 visual questions originating from blind people who each took a picture using a mobile phone and recorded a spoken question about it. Advertising is another field that considers the application of VQA. The application of VQA in the advertising field can help advertising designers to find the most successful ads. Ref. [10] proposes a theme recommender system for creative advertising strategists based on the VQA task. Ref. [11] is a survey that exhaustively investigates the different application fields of VQA.

We need to process various types of information in our daily lives, such as vision, sound, and taste. Multi-modal machine learning aims at processing and understanding multi-source modality information through machine learning. At present, the approaches to the VQA task can be classified into three main categories as follows. The first category uses a pure neural network (bilinear pooling [12], CNN, and so on) as the VQA structure for feature extraction, multi-modal fusion, and answer reasoning. The second category uses knowledge bases and databases as external information for answer reasoning [13,14]. The third category uses the neural-symbolic approach [15,16] to perform interpretable computations. This paper is interested in the multi-modal alignment using purely neural network approaches. The alignment of multi-modal information is responsible for finding the corresponding relationship between sub-branches of different modal information from the same instance.

The alignment of image and question in the VQA task is a challenging problem that decides which area of the image is most relevant to the given question. Inspired by human attention, the attention mechanism [17] is one of the most used methods in multi-modal alignment. With the attention mechanism, the model can selectively utilize given information. Many fruitful VQA models have utilized the attention mechanism, e.g., Bottom-Up and Top-Down (BUTD) [18] use bottom-up attention to align question information to each object feature. Dynamic Fusion with Intra and Inter-modality Attention Flow (DFAF) [19] and Deep Modular Co-Attention Networks (MCAN) [20] consider intra-attention within each modality and inter-attention across different modalities using the scaled dot-product attention from Transformer. When dealing with multi-modal feature fusion, many approaches simply use linear models to integrate the visual feature and textual feature, which ignores the correlation between multi-modal features. In consideration of the relationship between question words and image objects, Bilinear Attention Networks (BAN) [21] utilize the bilinear method [22,23] to fuse question and image information. The graph models [24,25] view the image as a graph network and build a connection between objects. The authors in [26] discussed a co-attention network from the perspective of graph neural networks.

However, when aligning two modalities with the self-attention mechanism, it only pays attention to the information of the current two objects. The neglected information of objects near these two objects can also influence the attention weight of the current two objects. The attention weight calculated in this way cannot fully capture the complex relationship between objects. Furthermore, there are always questions involving the position information of objects in the image. It is essential to capture the position information of each object in the image to answer such kinds of questions.

In this paper, we propose to build a multi-hop attention network to fully capture the information around two objects when calculating the attention weight between these two objects. Before multi-modal fusion, each word is concatenated with its adjacent word information, which makes full use of the information around a word when performing self-attention. We also embed the position information of each object into a vector and concatenate them with the feature of the objects. In this way, position information is taken into consideration, which enables the model to align question information with the relevant position of the image.

2. Materials and Methods

This section introduces several commonly used attention mechanisms and our VQA model in detail. The input question and image preprocessing are described after introducing the attention mechanism. Then, we provide the implementation details of our multi-hop attention layer. Last, we make full use of the position information of objects and build a position embedding to generate an attention distribution that considers the position information of objects in the image.

Figure 1 shows the main structure of our proposed model. The feature extractor extracts bounding boxes and object features of the image. The bounding boxes are converted into position embedding and concatenated to image features.

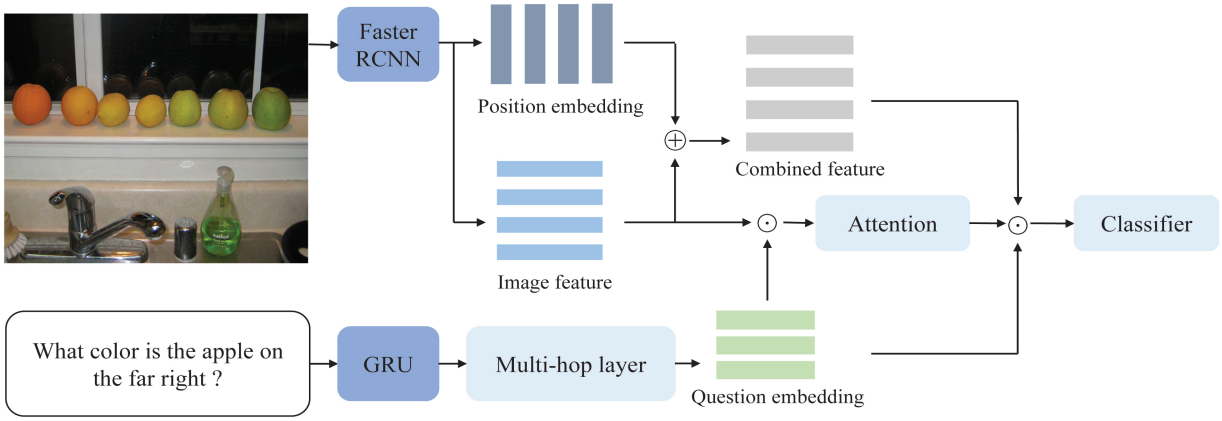


Figure 1. Overview of the proposed VQA model. $\oplus$ denotes concatenation operation and $\odot$ denotes matrix multiplication.

2.1. Classical Attention Mechanisms

The attention mechanism is widely used in many deep learning tasks, such as machine translation, speech recognition, and image captioning. Most of these attention layers are based on the encoder–decoder frame. In machine translation, the encoder encodes the input sentence and the attention layer distributes attention weights before decoding the sentence. Many VQA models can also be regarded as encoder–decoder models wherein the encoder encodes question–image pairs and the decoder outputs the answer.

2.1.1. Dot Product Attention

Dot product attention calculates the correlation between entities to build their interaction with each other. It converts the input into three new vectors, Key($(\boldsymbol{K})$), Query($(\boldsymbol{Q})$) and Value($(\boldsymbol{V})$), by multiplying the input to three random initialized matrices. The correlation between two entities is calculated by obtaining a dot product between Key and Query. The softmax function is used to obtain the attention weight. The attention weight is multiplied by the Value to build an attention distribution [17]:

$$\hat{\boldsymbol{V}} = softmax(\frac{\boldsymbol{K}\boldsymbol{Q}^T}{\frac{1}{\sqrt{d}}})\boldsymbol{V} \tag{1}$$

where $\hat{\boldsymbol{V}}$ is the updated value vector, $\frac{1}{\sqrt{d}}$ is the scaling factor, d denotes the dimension of Key, Query and Value.

2.1.2. Multi-Head Attention

To better capture information from input, multi-head attention uses n parallel dot product layers to calculate their attention weights. Each head can pay attention to a different part of the input. In these n parallel heads, each head has a Key Query dot product to

calculate the attention weight. Finally, the calculated Values are concatenated together and yield the final vector [17]:

$$att((\boldsymbol{K}, \boldsymbol{Q}), \boldsymbol{V}) = att((\boldsymbol{K}, \boldsymbol{Q}), \boldsymbol{v_1}); \ldots; att((\boldsymbol{K}, \boldsymbol{Q}), \boldsymbol{v_n}) \quad (2)$$

where att is the dot product to calculate the attention weight of each head. $att; att \ldots$ represents the concatenation of n parallel heads.

2.1.3. Multi-Modal Factorized Bilinear Pooling

Many tasks need to receive input information from two or more channels and discuss their relationship. The bilinear attention network was firstly designed to be implemented in this kind of task. In VQA, question and image are two kinds of information inputted from two channels. Classical bilinear pooling firstly uses a dot product to obtain an attention map between question and image features. The attention map is then multiplied by the two channels of question and image information [21]:

$$f(\boldsymbol{v}, \boldsymbol{q}) = \boldsymbol{v}^T \boldsymbol{W_i} \boldsymbol{q} \quad (3)$$

where $\boldsymbol{v}, \boldsymbol{q}$ denote the image feature and question vector. $\boldsymbol{W_i}$ is a projection matrix. Although bilinear pooling can find a connection between pairs of features in the multi-modal task, it also produces an enormous number of parameters and gives rise to a sharp increase in computational load. To reduce the rank of $\boldsymbol{W_i}$, the low-rank bilinear model is proposed. In the low-rank bilinear model, $\boldsymbol{W_i}$ is factorized as the multiplication of two low-rank matrices $\boldsymbol{U_i} \boldsymbol{V_i^T}$, where $\boldsymbol{U_i} \in \boldsymbol{R}^{N \times d}$ and $\boldsymbol{V_i} \in \boldsymbol{R}^{M \times d}$. In this way, the rank of $\boldsymbol{W_i}$ after replacement is at most $d \leq \min(N, M)$. For the scalar output a_i [21],

$$a_i = \boldsymbol{x}^T \boldsymbol{W_i} \boldsymbol{y} \approx \boldsymbol{x}^T \boldsymbol{U_i} \boldsymbol{V_i^T} y = \boldsymbol{1}^T (\boldsymbol{U_i^T} \boldsymbol{x} \circ \boldsymbol{V_i^T} \boldsymbol{y}) \quad (4)$$

where $\boldsymbol{1} \in \boldsymbol{R}^d$ is a vector of ones, and $\circ$ denotes the Hadamard product (element-wise multiplication) of two features.

For a vector output $\boldsymbol{f}$, a pooling matrix $\boldsymbol{P}$ is introduced [21]:

$$\boldsymbol{f} = \boldsymbol{P}(\boldsymbol{U}^T \boldsymbol{x} \circ \boldsymbol{V}^T \boldsymbol{y}) \quad (5)$$

where $\boldsymbol{P} \in \boldsymbol{R}^{c \times d}$, $\boldsymbol{U} \in \boldsymbol{R}^{N \times d}$ and $\boldsymbol{V} \in \boldsymbol{R}^{M \times d}$. It allows $\boldsymbol{U}$ and $\boldsymbol{V}$ to be two-dimensional tensors by introducing $\boldsymbol{P}$ for a vector output $\boldsymbol{f} \in \boldsymbol{R}^c$, significantly reducing the number of parameters.

Attention also provides an efficient mechanism to reduce the input channel by selectively utilizing given information. The attention weight β is defined by the output of the softmax function as [21]:

$$\beta = softmax(\boldsymbol{P}(\boldsymbol{U}^T \boldsymbol{x} \circ \boldsymbol{V}^T \boldsymbol{y})) \quad (6)$$

where $\boldsymbol{P} \in \boldsymbol{R}^{G \times d}$, $\boldsymbol{U} \in \boldsymbol{R}^{N \times d}$ and $\boldsymbol{V} \in \boldsymbol{R}^{M \times d}$. G represents the multiple glimpses mechanism. Finally, inputs $\boldsymbol{x}$ and $\boldsymbol{y}$ from two different channels can obtain the joint representation for the classifier to obtain the final answer. As shown in Figure 2, the m-dimensional input $\boldsymbol{x}$ and n-dimensional input $\boldsymbol{y}$ coming from two different channels are multiplied by two low-rank matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ separately. A Hadamard product is then used to combine the two features before calculating the attention weight via the softmax function.

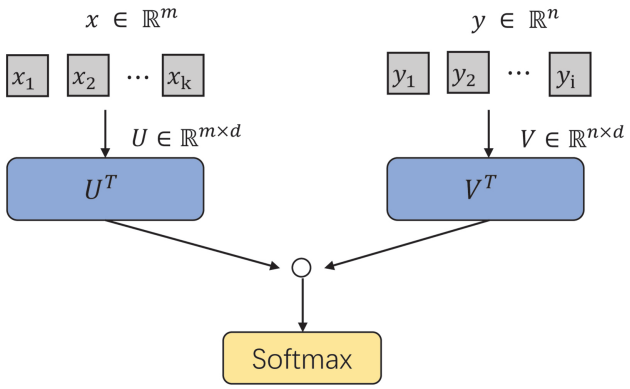


Figure 2. Flowchart of multi-modal factorized bilinear pooling module.

2.1.4. Top-Down Attention for VQA

Top-down attention is a classical question-guided attention mechanism common to many modern VQA models. In top-down attention, each feature vector $\boldsymbol{v_i}$ is concatenated with the question embedding $\boldsymbol{q}$. They are then passed through a non-linear layer and a linear layer to obtain a scalar attention weight α_i [18]:

$$\boldsymbol{a_i} = \boldsymbol{w} f(\boldsymbol{v_i}; \boldsymbol{q}) \tag{7}$$

$$\alpha = softmax(\boldsymbol{a}) \tag{8}$$

$$\hat{\boldsymbol{v}} = \sum_{i=1}^{K} \alpha_i \boldsymbol{v_i} \tag{9}$$

where $\boldsymbol{w}$ is a learned parameter vector. The attention weights are passed through a softmax function. The image features are then weighted by these attention weights and summed to obtain a single vector $\hat{\boldsymbol{v}}$ representing the attended image.

2.2. Model Input Preprocessing

The original input for an instance is a textual question and an image. We use two different networks to process information from the corresponding channels separately. For efficiency, the parameters of these networks are pretrained and held fixed during the training of the VQA model.

2.2.1. Question Word Embedding

For computational efficiency, the question length is fixed with a common length of 14 words, which makes the dimension of every question vector the same. The Global Vectors for Word Representation (GloVe) [27] word embedding is used to embed question words. The GloVe word embedding is pretrained on the Wikipedia corpus [28]. After embedding, each word is transformed into a vector with a common dimension of 300. To better learn the textual relationship of the question, the question vector after word embedding passes through a Gated Recurrent Unit (GRU) [29].

2.2.2. Image Feature Extraction

One hundred objects are extracted from each image using an object detector, corresponding to 100 object features. The number of objects extracted from the object detector will influence the final answer result. Extracting more objects from an image will increase the accuracy rate of the answer with the sacrifice of training speed. We use a pretrained Faster R-CNN [30] framework as the object detector, which can produce object-level features rather than a grid of features as produced by the traditional CNN [31,32]. After object detection, the 100 most related objects in the image are converted into 2048-dimensional

vectors. The bounding box is also extracted from the object detector as the positional information of the object.

2.3. Multi-Hop Attention Layer

In the self-attention mechanism used in previous VQA models, the calculation of attention weights depends solely on the information of the current two objects, which ignores information around these two objects. Inspired by the Multi-hop Attention Graph Neural Network (MAGNA) [33], in our model, we utilize a multi-hop attention layer to make full use of the information around an object before calculating attention weights.

Our multi-hop layer rebuilds a word vector by concatenating the following $n-1$ words to that word, which combines the information of n words to the same vector. In this way, every word vector in a question is compensated with word information around this word. The calculation of the attention weight between question words and image objects is then able to be viewed as a conditional probability between the current word and object, where the condition is surrounding word information. The last several words in a sentence that does not have $n-1$ words behind it will return to the beginning of the sentence and concatenate the words at the beginning. Word vectors after concatenation are passed through a non-linear layer with the ReLu [34] activation function. The concatenation of the ith word is defined as:

$$\boldsymbol{W_i} = f(\boldsymbol{w_i}; \boldsymbol{w_{i+1}}; \ldots; \boldsymbol{w_{i+n-1}}) \tag{10}$$

where $\boldsymbol{W_i}$ is the ith word vector after concatenation and $\boldsymbol{w_i}$ is the ith original word vector. f denotes the non-linear layer.

The question vector after the multi-hop layer is defined as $\boldsymbol{Q_n}$, where n represents the number of words concatenated to a vector. As mentioned above, we create several such kinds of question vectors with different multi-hop lengths from 1 to n. In consideration of the lower influence of words far away from the current words, these question vectors are then multiplied with a decay factor of $\frac{1}{d^2}$:

$$\boldsymbol{Q_n} = \frac{1}{d^2} \cdot (\boldsymbol{W_1}; \boldsymbol{W_2}; \ldots; \boldsymbol{W_k}) \tag{11}$$

where k is the number of words in a question (14 in our model). d is the multi-hop length (number of words concatenated together in a question word).

The new question vector $\boldsymbol{Q_n}$ obtained from the above layers is then used to calculate the attention map by obtaining the dot product with image features:

$$\boldsymbol{L_n} = (\boldsymbol{P} \circ \boldsymbol{V})^T \boldsymbol{Q_n} + \boldsymbol{b} \tag{12}$$

where $\boldsymbol{P}$, $\boldsymbol{b}$ are parameter matrices for training and $\circ$ denotes the Hadamard product. We perform the Hadamard product operation between image feature $\boldsymbol{V}$ and randomly initialize parameter matrix $\boldsymbol{P}$ before the dot product. The bias matrix $\boldsymbol{b}$ is also randomly initialized. $\boldsymbol{L_n}$ represents the attention map calculated by question vector $\boldsymbol{Q_n}$ and image feature $\boldsymbol{V}$. We calculate n numbers of attention maps $\boldsymbol{L_1}, \ldots, \boldsymbol{L_n}$ using question vectors with different multi-hop length $\boldsymbol{Q_1}, \ldots, \boldsymbol{Q_n}$. To integrate the joint representation from the multiple attention maps, we use a variant of a residual network to sum over every attention map. The integrated attention map is passed through a softmax function:

$$\boldsymbol{A} = softmax(\sum_{i=1}^{n} \theta_i \boldsymbol{L_i}) \tag{13}$$

where $\boldsymbol{A}$ is the final attention map output of the softmax function and θ_i is attention decay factor. Attention from different attention maps $\boldsymbol{L_1}, \ldots, \boldsymbol{L_n}$ is weighted differently by θ_i. In our implementation, we set $\theta_1 = 0.9$ and others to be 0.1, which turns out the best result. As shown in Figure 3, questions are firstly passed through a multi-hop layer and a non-

linear layer, which concatenates word vectors together and encodes them to new word vectors. The multi-hop layer concatenates different numbers of words from 1 to n, which corresponds to left to right in Figure 3. The re-encoded word vectors are multiplied by image features to obtain an attention distribution. The n numbers of attentions are added together (we compared different combination methods such as addition, concatenation, and multiplication, which indicates the add operation to be the best result) and passed through a softmax function, which obtains the overall attention distribution.

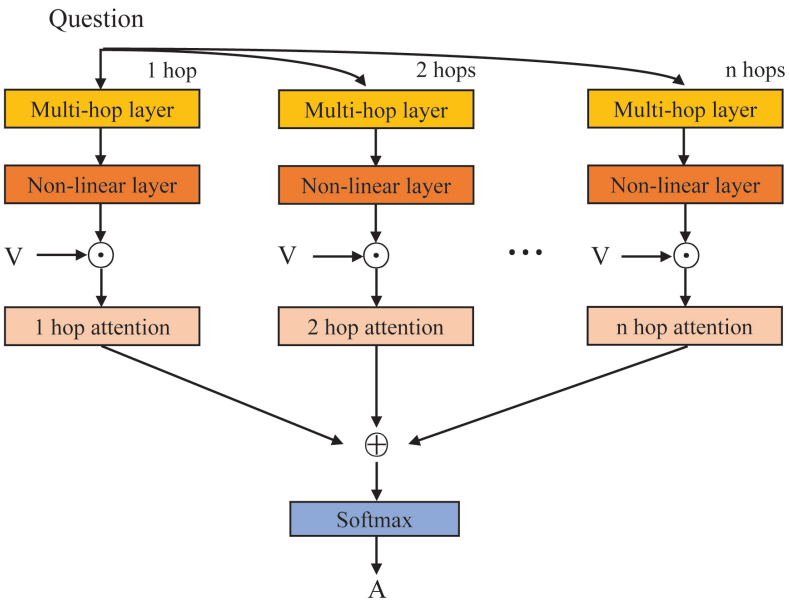


Figure 3. The structure of our multi-hop attention layer. $\oplus$ denotes add operation and $\odot$ denotes matrix multiplication.

2.4. Position Embedding Mechanism

In some situations, to answer a given question, the model needs to reason about the positional relationship between objects. As shown in Figure 4a, to answer the question "What is above the train?", the model needs to know the positional relationship between the train and the building. In this section, we build the position embedding layer to encode the position information of each object into a vector, which enables the model to align question information to the most relevant image objects in respect to the position.

The multi-modal alignment from the perspective of the object's position aims at choosing the positionally relevant objects in the image. We extract the upper-left and bottom-right coordinates of the bounding box as the position of objects. These coordinates are embedded into a vector $\boldsymbol{b_n}$. To facilitate subsequent processing, we normalize these coordinates by dividing them by the height and width of the image. The values of coordinates after normalization are in the range of [0, 1]. The distance for an object to the center of image $\boldsymbol{d_n}$ is calculated to show the general location of the object. The area $\boldsymbol{c_n}$ of an object's bounding box is also calculated to indicate the size relationship between objects. We concatenate this position information into vectors and pass it through a non-linear layer to obtain a position vector $\boldsymbol{P_n}$. The position vector $\boldsymbol{P_n}$ is then concatenated to image features:

$$\boldsymbol{P_n} = f(\boldsymbol{b_n}; \boldsymbol{d_n}; \boldsymbol{c_n}) \tag{14}$$

$$\hat{\boldsymbol{V}} = f(\boldsymbol{V}; \boldsymbol{P_n}) \tag{15}$$

where f is the non-linear layer and $\hat{\boldsymbol{V}}$ is the image features enriched by the position vector.

After the concatenation of the position vector, each object vector is composed of two parts: the object feature and the position information of this object. Those objects with the

most relevant feature and position information will be chosen to align with the question. Figure 4b shows the generation process of the position embedding vector $\boldsymbol{P_n}$.

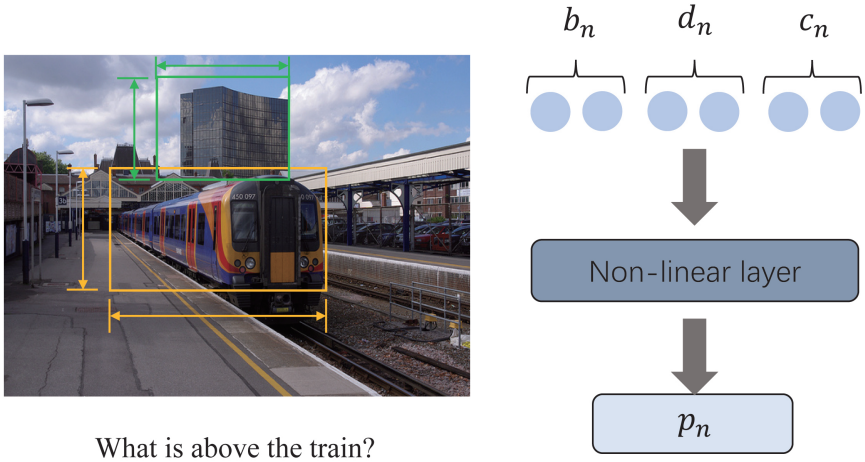


Figure 4. (**a**) An example of a question in relation to the positional relationship of objects. (**b**) Flowchart of the position embedding $\boldsymbol{P_n}$.

2.5. Multi-Modal Fusion and Classification

The attention map obtained in the above multi-hop layer is used to fuse question vectors and image features in the fusion layer. In the implementation, we use a bilinear attention network to fuse question and image:

$$\boldsymbol{h} = \hat{\boldsymbol{V}}^T \boldsymbol{W} \boldsymbol{q} \tag{16}$$

$$\boldsymbol{H} = \sigma(\boldsymbol{h}) \tag{17}$$

where $\hat{\boldsymbol{V}}$ is the image features concatenated with position information and $\boldsymbol{q}$ is the original question vector. $\boldsymbol{W}$ is the attention map obtained in the multi-hop layer. σ denotes a linear layer and $\boldsymbol{H}$ is the result after the fusion layer.

We view VQA as a classification task that sends a fused feature of question and image to a classifier and requires the classifier to output an answer for the given image feature and question feature. Here, we adopt the commonly used classification method, which selects 3129 answers that appear more than 8 times in the training set as the candidate answers. Each question in the VQA2.0 dataset corresponds to one or several answers with soft accuracy in [0, 1]. The classification layer in our VQA model is composed of a non-linear layer and a linear layer. The final feature vector after multi-modal fusion is passed through these two layers:

$$\boldsymbol{x} = f(\boldsymbol{H}) \tag{18}$$

$$\boldsymbol{X} = \sigma(\boldsymbol{x}) \tag{19}$$

where $\boldsymbol{X}$ is the output vocabulary of the answer. The output vocabulary is a 3129-dimensional vector, which corresponds to 3129 candidate answers.

3. Results

3.1. The Dataset

We evaluate our model on the VQA2.0 dataset [35], which is the updated version of the VQA dataset. It contains 204,721 COCO images, 1,105,904 questions, and 11,059,040 ground truth answers. The dataset is divided into training set, validation set, and test set. Every image in the dataset corresponds to several questions and every question has 10 soft

accuracy answers as annotations. Compared with the previous version, the VQA2.0 dataset reduces the answer bias by collecting complementary pairs.

The Visual Genome dataset [36] was released to promote research on the high-level semantic understanding of imagea. In this paper, it is used for data augmentation when training the model. The dataset contains 108 K images densely annotated with scene graphs containing objects, attributes, and relationships, as well as 1.7 M visual question answers. Images in the dataset are divided into several regions and each region has a sentence of natural language to describe the region.

We perform standard question text preprocessing and tokenization. Questions are held fixed to a length of 14 words for computational efficiency. We view VQA as a classification problem and candidate answers are selected from correct answers in the training set that appear more than 8 times, which turn out to be an output vocabulary size of 3129. Our VQA test server submissions are trained on the training and validation sets plus additional questions and answers from Visual Genome. To evaluate answer quality, we report accuracies using the standard VQA metric, which takes into account the occasional disagreement between annotators for the ground truth answers.

3.2. Implementation Details

The dimension of the original question vector passed through an LSTM layer is 1280 and it becomes $n \times 1280$ after it is passed through the multi-hop layer. The image features extracted by Faster R-CNN are 100 vectors with the dimension of 2048, corresponding to 100 objects in the image. Before calculating the attention weight, they are all passed through a non-linear layer, which transforms them into the same dimension of 3840 (3×1280). Every linear layer and non-linear layer is regularized by the dropout method [37,38] (the probability $p = 0.5$, except for the fusion layer with 0.2). The position vector passed through a non-linear layer has the dimension of 1280. The middle feature dimension in the fusion layer is 2×1280.

We use the Adamax optimizer to optimize our model. It is a variant of Adam [39]. The warm-up method is used when setting the learning rate. The learning rate is min $(0.5i \times 10^{-3}, 2 \times 10^{-3})$ at the first nine epochs, where i is the number of epochs. After nine epochs, the learning rate is decayed by $1/4$ with a decay step of 2. Due to the fact that there might exist multiple correct answers for a question, we utilize the binary cross-entropy loss (BCE) as the loss function:

$$L = \sum_{i=1}^{|A|} (y_i \log h(x_i) + (1 - y_i) \log (1 - h(x_i))) \tag{20}$$

where $y_i = \min(\frac{\text{number of people that provided answer } a_i}{3}, 1)$, and $h(x)$ is the sigmoid function [40,41].

VQA has always been viewed as a maximum likelihood estimation problem. The output of the classifier is a 3129-dimensional vector and the annotation is also a 3129-dimensional vector, where 3129 represents the candidate answers that appear more than eight times in the training set. The VQA evaluation metric considers inter-human variability, defined as [42]:

$$Accuracy(ans) = \min(\text{humans that said ans}/3, 1) \tag{21}$$

We use one NVIDIA GeForce RTX3070 GPU for the experiment, which has 5888 CUDA cores, 1.5 GHz GPU frequency, 256-bit memory interface width, G6 graphical memory, 8 GB graphical memory amount, and 16 GHz graphical memory frequency. The CPU name is Intel i5 10,600 kf and the main memory amount is 32 GB. To save the memory of the GPU, the batch size of our model is set to 64 and the epoch time is set to 20 times.

3.3. Ablation Study

We conduct several ablation studies to verify the effectiveness of each module in our network. Each experiment is repeated three times, training the same network with different random seeds. Table 1 shows the accuracy of our model on the VQA v2 dataset when utilizing different modules. When training each module, the setting of other modules is held fixed. Multi-hop refers to the multi-hop layer and n is the multi-hop length (number of words concatenated together in a question word). As shown in Table 1, the value of the decay factor for the question vector can influence the resulting accuracy. We set different n numbers from 1 to 4 to explore the accuracy of the model when using different multi-hop lengths. When we set $n = 2$, the result is much better than using the original question vector ($n = 1$). However, the accuracy drops when $n > 2$, which indicates that simply adding the multi-hop length could not improve the model accuracy. Based on the multi-hop module, we add a position embedding module to explore the effectiveness of position information. The result in Table 1 shows that our model can achieve an obvious improvement after adding position embedding.

Table 1. Ablation studies of the decay factor, multi-hop layer, and the position embedding. After the model adds position embedding, we verify the model accuracy with the change in the position embedding dimension.

Module	Setting	Accuracy
Decay factor	d = 1	65.23
	d = 2	65.56
	d = 3	65.42
Multi-hop	n = 1	64.96
	n = 2	65.56
	n = 3	65.33
	n = 4	65.07
Multi-hop + position	Position dimension = 512	64.86
	Position dimension = 1024	65.54
	Position dimension = 1280	65.77

3.4. Comparison with Other Models

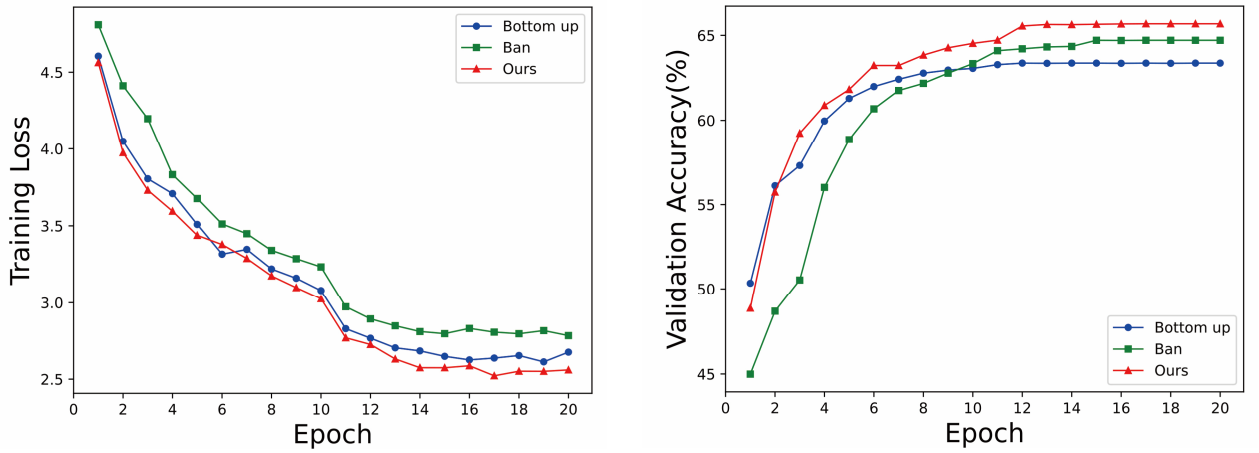
In this section, we compare our model with some other classical models to better show the effectiveness of our model.

Table 2 shows the performance of our proposed model and state-of-the-art methods trained on VQA. LV_NUS [43] offers a novel loss function for VQA that more closely reflects a VQA model's performance. Bottom-Up [18] is the winner of the VQA 2017 challenge, and it first applies detected object features instead of grid features. It proposes a combined bottom-up and top-down attention mechanism that enables attention to be calculated at the level of objects. The Dense Co-Attention Network (DCN) [44] utilized a dense stack of multiple layers of a co-attention mechanism that significantly outperforms previous methods with ResNet features. MFB [45] develops a Multi-modal Factorized Bilinear Pooling approach to fuse the visual features from images with the textual features from questions and uses a co-attention learning architecture to jointly learn both image and question attention. BAN [21] extends the idea of co-attention into bilinear attention, which considers every pair of multi-modal channels, such as the pairs of question words and image regions. It introduces a bilinear attention map to reduce the input channel of both modalities. It applies a variant of residual learning and a multi-glimpses method to further increase the model's accuracy, with a sacrifice in terms of computational complexity.

Table 2. Comparison of our model with several previous state-of-the-art models.

Model	Batch Size	Accuracy
LV_NUS [43]	64	60.4
MFB [45]	64	60.9
DCN [44]	64	62.94
Bottom-up [18]	64	63.36
BAN-1 [21]	64	65.01
Ours	64	65.77

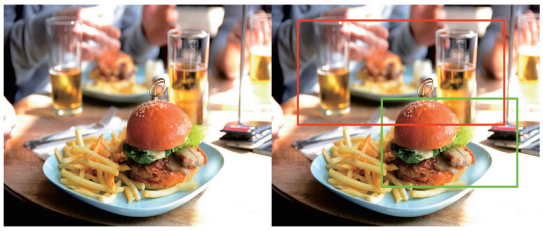
Figure 5 shows the training loss and validation accuracy of each epoch. As shown in the figure, our model tends to converge after the 12th epoch and the prediction accuracy becomes stable. However, the training loss still drops after accuracy stops rising.



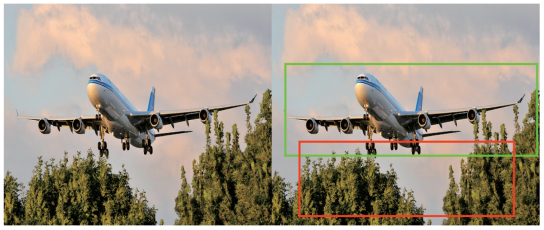
(**a**) (**b**)

Figure 5. (**a**) The training loss of our model vs. epoch of Bottom-Up and Ban; (**b**) The validation accuracy of our model vs. epoch of Bottom-Up and Ban.

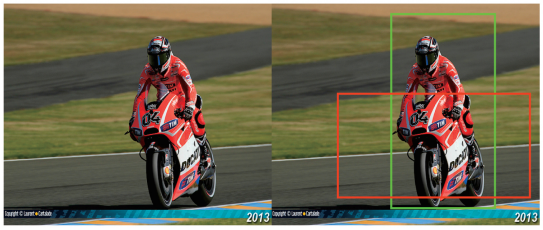
Some results of our model are visualized in Figure 6, which were randomly selected from the validation set. A is the annotation of every image question pair and P is the prediction answer given by our model. The two examples in the first row give correct answers, while the second row gives incorrect answers. To answer questions in the first row, the model needs to locate the significant object in the image. These two examples show the significance of our position embedding mechanism. When answering questions with high-level reasoning as in the third example, or questions related to the color of objects as in the fourth example, the model cannot perform well with regard to position information.



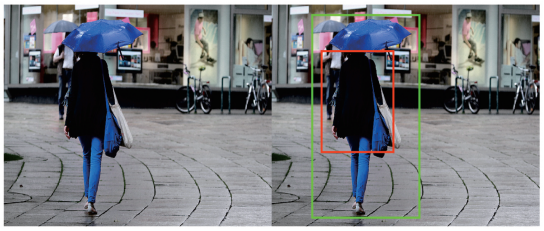
(1) Is there a big burger on the plate?
A: Yes
P: Yes



(2) Is the plane going to hit the trees?
A: No
P: No



(3) Is the bike going left?
A: Yes
P: No



(4) What color is the woman's pants?
A: Blue
P: Black

Figure 6. Several examples of the reasoning results of our VQA model. Different colors of the bounding boxes denote the different objects detected.

4. Conclusions

In this paper, we propose a multi-hop attention alignment method to provide more information when aligning two modalities of image and question. Our approach enables attention to be calculated in the context of information about surrounding objects. The mapping of question words to image objects is highly enriched in the VQA task. To consider position information, the position embedding mechanism is proposed. Applying this approach to answer reasoning can enable the model to better understand objects' position relationships. We believe that our proposed methods provides an idea for learning relational concepts of visual reasoning.

Author Contributions: The authors confirm their contributions to the paper as follows: Conceptualization, Q.X. and C.Y.; methodology, Q.X.; software, C.Y.; validation, Y.H. and P.P.; formal analysis, Y.H.; investigation, C.Y. and P.P.; resources, Z.Z. and W.C; data curation, Q.X.; writing—original draft preparation, Q.X. and C.Y.; visualization, Y.H. and P.P.; supervision, Z.Z. and W.C.; project administration, Y.H. and C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was sponsored by the National Natural Science Foundation of China (No. 61771197).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request from the authors.

Acknowledgments: The authors would thank the providers of the VQA2.0 dataset for their contribution to the VQA task and our experiment.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

VQA	Visual Question Answering
BUTD	Bottom-Up and Top-Down
GRU	Gated Recurrent Unit
BAN	Bilinear Attention Networks
BGN	Bilinear Graph Networks
GloVe	Global Vectors for Word Representation
MAGNA	Multi-hop Attention Graph Neural Network

References

1. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R. Show, attend and tell: neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
2. Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C.L.; Parikh, D. Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 2425–2433.
3. Das, A.; Kottur, S.; Gupta, K.; Singh, A.; Yadav, D.; Moura, J.M.F.; Parikh, D.; Batra, D. Visual dialog. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
4. Guo, D.; Xu, C.; Tao, D. Image-question-answer synergistic network for visual dialog. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019.
5. Lau, J.J.; Gayen, S.; Abacha, A.B.; Demner-Fushman, D. A dataset of clinically generated visual questions and answers about radiology images. *Sci. Data* **2018**, *5*, 1–10.
6. He, X.; Zhang, Y.; Mou, L.; Xing, E.; Xie, P. PathVQA: 30,000+ Questions for Medical Visual Question Answering. *arXiv* **2020**, arXiv:2003.10286. [CrossRef] [PubMed]
7. Vu, M.H.; Löfstedt, T.; Nyholm, T.; Sznitman, R. A Question-Centric Model for Visual Question Answering in Medical Imaging. *IEEE Trans. Med. Imaging* **2020**, *39*, 2856–2868.
8. Ren, F.; Zhou, Y. CGMVQA: A New Classification and Generative Model for Medical Visual Question Answering. *IEEE Access* **2020**, *8*, 50626–50636. [CrossRef] [PubMed]
9. Gurari, D.; Li, Q.; Stangl, A.J.; Guo, A.; Lin, C.; Grauman, K.; Luo, J.; Bigham, J.P. VizWiz Grand Challenge: Answering Visual Questions From Blind People. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3608–3617. [CrossRef]
10. Zhou, Y.; Mishra, S.; Verma, M.; Bhamidipati, N.; Wang, W. Recommending themes for ad creative design via visual-linguistic representations. In Proceedings of the Web Conference, 2020, Taipei Taiwan, 20–24 April 2020; pp. 2521–2527.
11. Barra, S.; Bisogni, C.; Marsico, M.D.; Ricciardi, S. Visual question answering: Which investigated applications? *Pattern Recognit. Lett.* **2021**, *151*, 325–331.
12. Fukui, A.; Park, D.H.; Yang, D.; Rohrbach, A.; Darrell, T.; Rohrbach, M. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. *arXiv* **2016**, arXiv:1606.01847. [CrossRef]
13. Vo, H.Q.; Phung, T.; Ly, N.Q. VQASTO: Visual question answering system for action surveillance based on task ontology. In Proceedings of the 2020 7th NAFOSTED Conference on Information and Computer Science, Ho Chi Minh City, Vietnam, 26–27 November 2020; pp. 273–279.
14. Yu, J.; Zhu, Z.; Wang, Y.; Zhang, W.; Hu, Y.; Tan, J. Cross-modal knowledge reasoning for knowledge-based visual question answering. *Pattern Recognit.* **2020**, *108*, 107563.
15. Mao, J.; Gan, C.; Kohli, P.; Tenenbaum, J.B.; Wu, J. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv* **2019**, arXiv:1904.12584. [CrossRef]
16. Kovalev, A.K.; Shaban, M.; Osipov, E.; Panov, A.I. Vector Semiotic Model for Visual Question Answering. *Cogn. Syst. Res.* **2022**, *71*, 52–63.
17. Vaswani, A.; Shazeerer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. [CrossRef]
18. Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; Zhang, L. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6077–6086.
19. Gao, P.; Jiang, Z.; You, H.; Lu, P.; Hoi, S.C.H.; Wang, X.; Li, H. Dynamic Fusion With Intra- and Inter-Modality Attention Flow for Visual Question Answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
20. Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; Tian, Q. Deep Modular Co-Attention Networks for Visual Question Answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
21. Kim, J.; Jun, J.; Zhang, B. Bilinear Attention Networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–13.
22. Kim, J.; On, K.; Lim, W.; Kim, J.; Ha, J.; Zhang, B. Hadamard Product for Low-rank Bilinear Pooling. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

23. Pirsiavash, H.; Ramanan, D.; Fowlkes, C. Bilinear classifiers for visual recognition. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 1–9.
24. Norcliffe-Brown, W.; Vafeias, E.; Parisot, S. Learning Conditioned Graph Structures for Interpretable Visual Question Answering. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–10.
25. Hu, R.; Rohrbach, A.; Darrell, T.; Saenko, K. Language-Conditioned Graph Networks for Relational Reasoning. In Proceedings of the International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
26. Guo, D.; Xu, C.; Tao, D. Bilinear Graph Networks for Visual Question Answering. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–12.
27. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014. [CrossRef] [PubMed]
28. Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. Available online: http://nlp.stanford.edu/projects/glove/ (accessed on 1 March 2022).
29. Cho, K.; Merriënboer, B.V.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the European Conference on Computer Vision, Las Vegas, NV, USA, 27–30 June 2016.
33. Wang, G.; Ying, R.; Huang, J.; Leskovec, J. Multi-hop Attention Graph Neural Network. *arXiv* **2020**, arXiv:2009.14332.
34. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
35. Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; Parikh, D. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
36. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D.A.; et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73.
37. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958. . [CrossRef]
38. Shen, X.; Tian, X.; Liu, T.; Xu, F.; Tao, D. Continuous Dropout. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3926–3937.
39. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
40. Han, J.; Moraga, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. *Nat. Artif. Neural Comput.* **1995**, *930*, 195–201. [CrossRef]
41. Yin, X.; Goudriaan, J.; Lantinga, E.A.; Vos, J.; Spiertz, H.J. A Flexible Sigmoid Function of Determinate Growth. *Ann. Bot.* **2003**, *91*, 361–371.
42. VQA: Visual Question Answering. Available online: https://visualqa.org/ (accessed on 1 March 2022).
43. Ilievski, I.; Feng, J. A Simple Loss Function for Improving the Convergence and Accuracy of Visual Question Answering Models. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef] [PubMed]
44. Nguyen, D.; Okatani, T. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6087–6096.
45. Yu, Z.; Yu, J.; Fan, J.; Tao, D. Multi-Modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering. In Proceedings of the IEEE International Conference on Computer Vision (ICCV),Piscataway, NJ, USA, 22–29 October 2017; pp. 1821–1830.