



Multiple answers to a question: a new approach for visual question answering

Sayedshayan Hashemi Hosseinabad¹ · Mehran Safayani¹ · Abdolreza Mirzaei¹

Published online: 1 January 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

With the advent of deep learning, multi-modal data have been of great interest. One of the multi-modal tasks which can be included in the computer vision domain is visual question answering (VQA). In VQA, a question and an image are entered into the model and the model tries to answer the question according to the image. To the best of our knowledge, the current techniques look at the image and only give one answer to the question asked. However, in some situations, there are several answers to the asked question. In this paper, we address this problem and define a new domain in the task of VQA as well as a new computationally efficient approach to cope with multiple-answer VQA. In this approach, we use a sliding window in an efficient manner to examine the answer to the question in different parts of the image. Due to the fact that so far no proper dataset is available for multiple-answer VQA, we provide a new dataset for evaluating our proposed model. The experiments express that our model uses 94% less operation than other models, making it very suitable for real-time applications.

Keywords Visual question answering · Deep learning · Convolution neural network · Multiple answers · Recurrent neural network

1 Introduction

As a member of the broad family of machine learning, deep learning has proven itself as a viable solution for many artificial intelligence problems. Deep neural networks outperform other machine learning solutions in many tasks. They can classify images [9], track objects in a video [2,4], analyze crowd behavior [25], crop images [27] or even choose hyper-parameter for another deep neural network [3] with a reasonable amount of accuracy. Although deep neural networks have numerous use cases, what is concentrated through this paper is a particular problem. We want to discuss the problem of visual question answering (VQA).

VQA has been around for a couple of years, and various applications have been mentioned for it such as answering

general questions about blind peoples' visual surroundings [11]. It also has very thin borders with the task of image captioning [13]. The main job of a VQA model is to answer a question asked from an image. So the inputs are a question and an image, and the output is a vector where each element indicates the probability of each answer. The first VQA models were proposed just in order to address the problem of the multi-modal classification task [17,19,31]. Later on, the models became very complex [5,15], for instance, some of them retrieving an attention mask for the image based on the question [14,22,30]. On the other hand, at present, the size of the VQA datasets is much larger than when VQA was introduced. The first datasets contained about one thousand images, while later released datasets contain more than two hundred thousand images [6].

All these amazing works have played a significant role in the development of VQA. However, there has always been a blind spot since the beginning. In all of the previous works, to the best of our knowledge, the inputs to the model are an image and a question, while the answer is yielded by processing the image as a whole. Although this might be an interesting research problem, the assumption of the existence of only one answer in a real-world image is nothing but an oversimplification of the problem. The image may con-

✉ Mehran Safayani
safayani@cc.iut.ac.ir

Sayedshayan Hashemi Hosseinabad
shayan.hashemi@ec.iut.ac.ir

Abdolreza Mirzaei
mirzaei@cc.iut.ac.ir

¹ Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran



(a) What color is the banana? (b) Is the child happy?

Fig. 1 Samples of multi-answer questions

tain multiple answers in different regions which has not been considered in the current models. For example, in Fig. 1a, we expect that the model detects multiple answers and produces an output vector which indicates yellow, green and black. We also expect that our model outputs both “yes” and “no” in Fig. 1b. These images are two samples of the existence of multiple answers in a real-world image. Moreover, in the example of answering general questions about blinds people’s visual surroundings whenever the blind person asks a question about whom is talking to, the system should prepare the answers with respect to the all people available in the scene. In order to tackle this, we define a new type of VQA problem which includes multiple answers in the image for the question. Our solution which applies to the certain types of questions is to cut the image into the glimpses (small sub-images) where each one contains only one answer to the question and train the model using those small images. In the test time, we slide over the image and each window of the image is fed into the model with a common question. Finally, the outputs of the system are aggregated and the final result is obtained.

Training on the small glimpses may have several advantages. One is the fact that the model does not need to be trained using the unnecessary details of the image. For example, in an office scene, none of the objects should affect the answer to a question about the color of the chair, but the chair itself. This is due to the fact that the image contains more than enough details required to answer the questions. Another advantage is that in the test time, we are able to use an image of any size since we chunk the image into the glimpses. Therefore, we can pass as many glimpses as we want into the model and aggregate the results later. This fact enables us to process multiple scales of the image and not missing an answer because of an object being too small or too big for our window size. However, everything comes with a price, although it requires less computational power, fewer resources and train faster during the training time, the number of operations required in the test time is massive. In this paper, we want to address this problem in a new way inspired by Sermanet [21] and keep the computational cost as low as possible. Our model architecture consists of a convo-

lutional neural network (CNN) for extracting features from each glimpse as well as a long short-term memory (LSTM) network for embedding the question. However, we produce all the answers simultaneous through a spatial output map. As our model has very low computational cost, it can be fit into real-time applications.

For assessing our method, we need a multi-answer dataset. By the time of writing this paper, we cannot find a proper dataset for this task. So we introduce a new dataset for VQA designed to have several answers in an image for a question. Our main contributions can be summarized as follows:

- Introducing multi-answer problem for the task of VQA.
- Proposing a new model to address the multi-answer VQA using the sliding window in a computationally efficient way.
- Preparing a new dataset named “ICQA” to train and evaluate our new sliding window approach.

The remainder of this paper is organized as follows: Some previous works are reviewed in Sect. 2; in Sect. 3, our approach and models are discussed; all the experiments are presented in Sect. 4; we have a discussion about our approaches in Sect. 5; the article is concluded in Sect. 6.

2 Related works

The models for the task of VQA are grouped into two categories as discussed by Gupta [7]. The first category is the models which look to all the pixels of the image with the same amount of weight and importance. The second group called attention-based models are those which find a way to weight some pixels more than others and then make decisions based on the weight of the pixels.

In non-attention models, the network takes two inputs (question and image) and embeds them into two vectors by choosing a favorite trainable method of deep learning such as CNN [12], LSTM [8], gated recurrent unit (GRU) [1] and fully connected layer. Then, each proposed model tries to combine two feature vectors in its own way and pass it to the classification layer (usually a fully connected layer with softmax activation) that will decide on what the model should choose as its output (since each output is an answer).

One of the baseline models for visual question answering is iBOWIMG as described by Zhou [31], and this model embeds the image using the pre-trained VGGnet [23] and the question using the bag of word method. After both inputs are embedded, iBOWIMG concatenates the embedded vectors and enters them to a fully connected layer with the softmax activation. Being a simple model, iBOWIMG gets quite good results.

As mentioned by Malinowski [17], Ask Your Neurons (AYN) approach is as interesting as its name. First, it tries to encode the image using VGGnet. After that, on each timestep of the question, the VGGnet extracted features are entered into an LSTM module along with the corresponding word, and the output of the LSTM at the latest timestep is fed to a fully connected layer with a softmax activation to predict the output. On the other hand, Vis+LSTM explained by Ren [19], which is a modified version of AYN, enters the image as the first word (and the last word in the bidirectional version) of the question sequence to the LSTM. The rest of Vis+LSTM architecture matches with AYN.

One of the models with the highest accuracy is discussed by Ma [15]. The model is known as Full CNN; looking at the name, it can be guessed that the model utilizes convolutional layers only. Full CNN extracts features from question and image using a one-dimensional convolutional neural network and a VGGnet like network architecture. Feature vectors are merged into one vector and passed into a new convolutional layer called multi-modal CNN, and the result of that layer is passed to a fully connected layer with the softmax activation.

Instead of concatenating, merging or any sort of vector combination, [5] calculates the correlation matrix between the question and the image feature vectors and hands its elements over to a fully connected layer. It is clear that calculating a huge correlation matrix is not achievable due to the memory limitation so the authors try to create a shortcut to calculate the correlation matrix and the fully connected layer altogether using the fast Fourier transformation. Another marvelous model in the category of non-attention models is dynamic parameter prediction or DPPnet described by Noh [18]. DPPnet uses VGGnet to extract the features from the image and puts a couple of fully connected layers on top of that for the classification. However, the difference between DPPnet and an image classification network is that the parameters of a fully connected layer are not trainable and are set by another network that gets the question as input and process it using a GRU. The innovative solution of DPPnet has made it one of the most accurate non-attention models.

Unlike non-attention models, attention-based models do not look at the whole image to do the classification, and there is a mechanism to find attention corresponding to a pixel or a region of the input image. After collecting the attention, each pixel or region will be multiplied by its corresponding attention. Then, we have an image that is more colorful in the important areas and almost black in the non-important areas. Before exploring the attention-based popular models, it is worth mentioning that attention is not exclusive to VQA. Other tasks could also take advantage of the attention-based approach to boost their performance. For example, attention is used in the context of visual saliency [26,28,29].

“Where to Look (WTL)” was one of the first attention-based models proposed by Shih [22]. At first, it transforms the feature vectors of each region of the image and the question into the space with the same size using two separate linear transformations. By computing the inner product of the two transformed vectors, the attention of each region is obtained. This operation is repeated until the all regions’ attentions are computed. The final image representation is the average of the different regions weighted by the corresponding attention. Thereafter, the image and the question representations are concatenated and fed to two fully connected layers following by a softmax to be classified.

A slight modification to WTL has made stacked attention network, which is found by Yang [30]. It can also be called one of the best models of VQA. SAN uses [10,23,24] to embed the input image. Its main idea is to create multiple layers of attention computation on top of each other due to the complexity of the question. According to SAN, some questions cannot be attended in one step and need to be attended in more steps. Feeding forward the attention applied image to the next attention module will cause a stack of attention matrixes to be applied to the image. The approach led to a much more accurate attention matrix.

A very complex and different approach for finding the attention is discussed by Lu [14]. Hierarchical co-attention obtains attention in two ways: the first one called “parallel co-attention”, where the image and the question attend to each other at the same time; the second called “alternating co-attention” that computes the correlation between the image and the question.

3 Our proposed methods for multi-answer VQA

All models discussed in the previous section have done a noble job addressing the problem of visual question answering. As mentioned earlier, to the best of our knowledge, none of the previous works have addressed the problem of multi-answer VQA. We introduce a new training approach as well as a new model architecture to address this problem.

3.1 Glimpse training approach

We want to discuss a new training approach which its main idea is to train the network on proper and less detailed glimpses of the image which does not contain more than one answer to the question rather than the full detailed images with multiple answers. For instance, to cope with image like Fig. 1a, we should provide images of bananas with different colors as the training data to the model. These images can be achieved by cropping the large images in order to have only one answer to the asked question. This training approach is

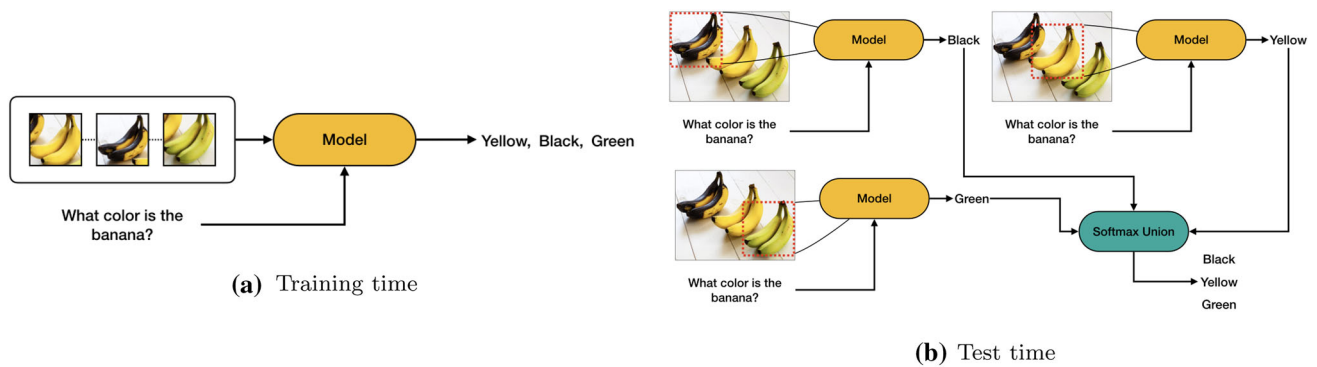


Fig. 2 Visualization of inputs and outputs of the glimpse training approach

illustrated in Fig. 2a. It is anticipated that the model can learn to answer certain types of questions more accurately and better results in the training can be achieved. We called this approach the “glimpse training” approach, as it trains with glimpses of the large images. It is worth mentioning that the glimpse training approach is designed to solve specific types of questions (at its current state). It cannot answer questions that associate with location or count.

In the test time, the images are larger than the glimpses. So we take a window with the same size of the training images, slide it over the large image and compute the result for each window along with the question.

The sliding window approach gives us multiple results each one coming from a specific location of the image. We can say the output is a two-dimensional tensor that its first dimension is representing the sliding window location’s index and the second one is the representation of the probability of each choice in that location. The target is a one-dimensional vector, where each element represents the independent probability of a single choice. In order to get the answers, we need to aggregate the two-dimensional tensor into a vector of independent probabilities. For this purpose, we take the maximum of it along with the first dimension, and we call this operation “SoftmaxUnion” which can be denoted as:

$$\text{SoftmaxUnion}(T) = \max_{p \in 1 \dots P} (T[p, :]),$$

where T is the two-dimensional output tensor of the sliding window approach and P is the number of all of the sliding window locations (with respect to its size and stride). The sliding window approach is visualized in Fig. 2b. After computing the output of the *SoftmaxUnion*, we pick all of the choices which pass a certain threshold as the final answers.

It is obvious that all the mentioned models can use this approach in order to be able to address the multi-answer problem.

3.2 Efficient multi-answer VQA (EMQA)

Glimpse training approach helps us to solve the multi-answer problem using much fewer data. In the test time, we should slide the classifier window over the large test images. This operation has very high computational cost. This problem is due to the fact that nearby windows have overlap with each other and most of the operations are the same.

In order to resolve this problem, we design a new architecture inspired by Sermanet [21] which proposed an integrated framework build on top of the convolutional neural network for classification, localization and detection. It developed a way of implementing the sliding window approach efficiently. However, it works only with one modality (image), while in the task of VQA, there are two modalities (image and text). To tackle this problem, we develop a model with the following constraints:

- Only convolutional layers are allowed, fully connected layers are implemented using convolutional layers with the kernel size of 1×1 .
- The strides size on the large image is equal to the product of strides of all layers in the network.
- Each layer’s strides must be chosen so that all of the convolution steps are complete for both glimpse and large images. In other words, if the input, kernel and stride sizes are $l \times l$, $k \times k$ and s , respectively, the following equation must be satisfied:

$$(l - k) \bmod s = 0.$$

- Paddings are not allowed in any layer.
- The images network for glimpse training should be designed so that its output is a tensor of size $1 \times 1 \times C$, where C is the number of classes.

Designing a classifier with these constraint will make the model robust to the input size variations, considering that the number of parameters in each layer of convolution is

Table 1 Fully convolutional network specifications

Layer specifications	First layer	Second layer	Third layer	Fourth layer	Fifth layer	Sixth layer	Seventh layer	Eighth layer
Channels	32	128	256	512	1024	512	256	123
Kernel size	7×7	5×5	4×4	2×2	2×2	2×2	2×2	2×2
Strides	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(1, 1)	(1, 1)	(1, 1)	(1, 1)
Padding	No	No	No	No	No	No	No	No
Max pooling	No	No	No	No	No	No	No	No
Glimpse image output size	29×29	13×13	5×5	2×2	1×1	1×1	1×1	1×1
Large image output size	253×253	125×125	61×61	30×30	29×29	29×29	29×29	29×29

not dependent on the input size of the layer. The only thing that changes with the input size in a convolutional layer is the output of that layer which is an input to another convolutional layer. When the model’s input is scaled up, the output will be a spatial tensor, indicating the probability of each class in spatial position in the input image. Iterating over the first two dimensions will give us the answer for each glimpse of the image associated with the corresponding region.

To apply this idea, the problem we face is that the task of VQA is not a singular modality task. We have to somehow affect the network by the embedded question without breaking the constraints of a fully convolutional network. In order to solve this problem, we embed the question using an LSTM layer and get a single vector that represents the question. Then, we choose last layer of convolution in our fully convolutional network and concatenate the question representation vector to the image channels of every region of the input tensor of that layer. Then, the output of this layer is fed into two 1×1 convolution layers. The 1×1 convolutions not only allow us to decrease the number of input channels but also allow the network to compute an output based on both image and question. The number of 1×1 layers of convolution might vary based on the dataset or the task. The output of the EMQA is a three-dimensional tensor. The first two dimensions are indicating the position, while the third one is the probability of each choice in that position. In this regard, If the small image and the large images have the pixel sizes of $l \times l$ and $L \times L$, respectively, and the stride of layer i is s_i , then the spatial output of the network is $o \times o$ where o is obtained by:

$$o = \frac{L - l}{\prod_i s_i} + 1.$$

Therefore, to obtain the proper output dimensions, the strides of each layer in the network should be designed carefully. Table 1 describes the detailed specification of each layer, while Fig. 3 shows a visualization of the model during the training and test time.

In order to aggregate the output of the model, we use the *SoftmaxUnion* but the output of the EMQA is a three-dimensional tensor. So we redefine

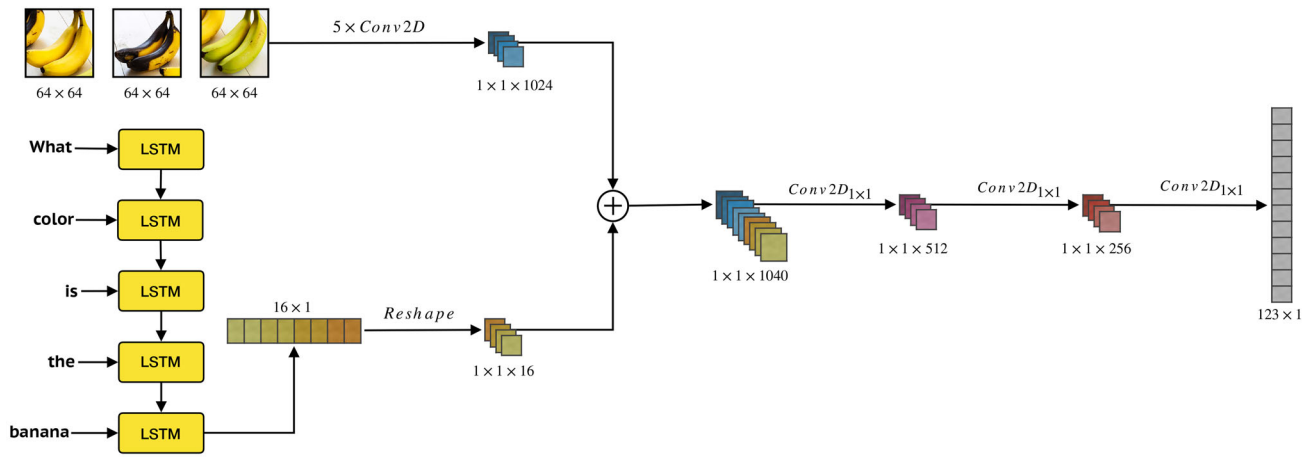
$$SoftmaxUnion(T) = \max_{w \in 0 \dots W, h \in 0 \dots H} (T[w, h, :]),$$

where T is the output tensor, W is the width of the output tensor and H is the height of the output tensor. After computing the output of the *SoftmaxUnion*, we pick all of the choices which pass a certain threshold as the final answers. As it is expected, the number of floating point operations (flops) decreases dramatically in the test time. It is worth mentioning that by increasing the input image size the difference in the computational cost between this approach and the sliding window approach increases.

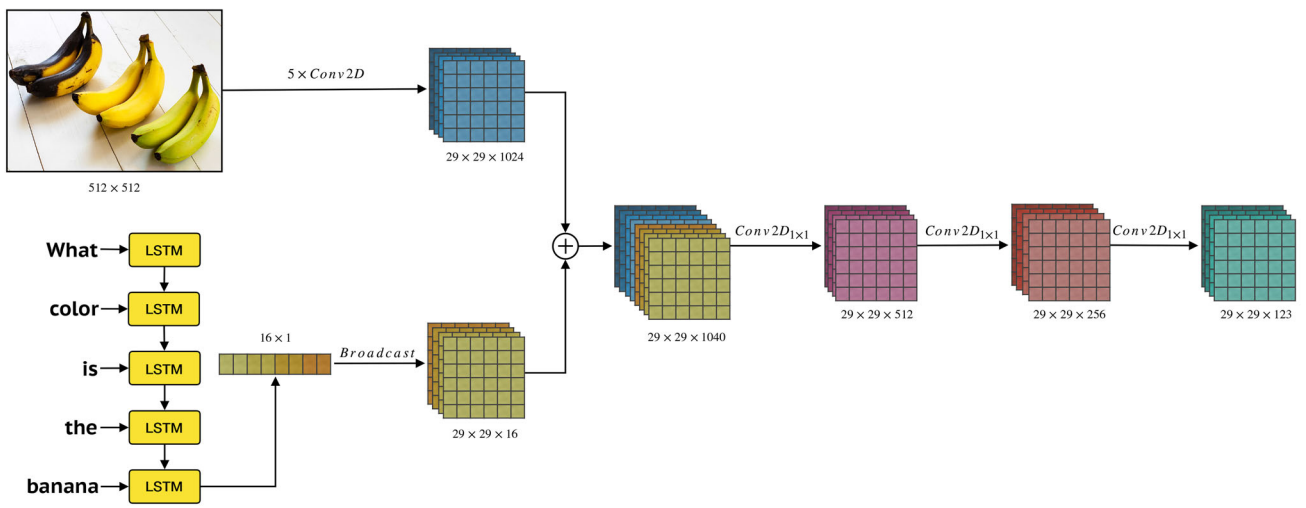
The goal of EMQA is to design an architecture that reduces the computational cost of the sliding window approach. Although approaches such as attention computation [14,26,29,30] or model complication [18] can achieve higher accuracy than EMQA due to their complex architectures, they have higher computation cost so that they cannot be used in real-time problem.

4 Experiments

In this section, we evaluate our approaches in terms of accuracy and computational efficiency. Our model is supposed to output all possible answers by inputting an image and a question. Since we did not find an appropriate dataset for this task and other datasets did not contain multiple answers for a question in an image, we decided to create a dataset for the multi-answer VQA from the ground up which will be explained in detail in the next subsection. In Sect. 4.2, we evaluate the glimpse training approach, while Sect. 4.3 introduces a method of visualization for glimpse training approach. The bias modality test of EMQA is demonstrated in Sect. 4.4. Section 4.5 suggests and evaluates a method for too large or too small objects. Section 4.6 discusses another



(a) EMQA visualization during the training time



(b) EMQA visualization during the test time

Fig. 3 Visualization of EMQA

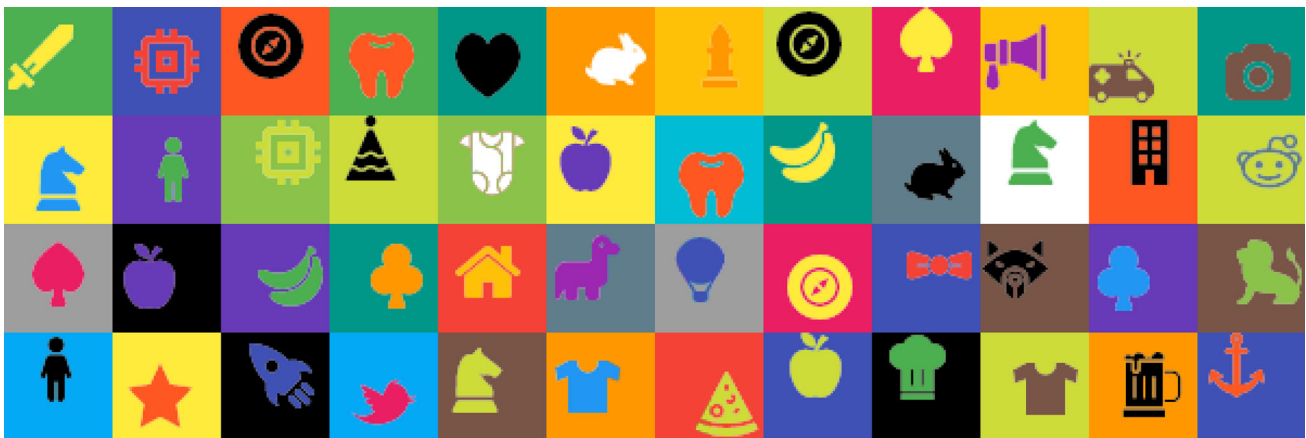


Fig. 4 Some samples of iconic shape used in the ICQA dataset

method for the multi-answer problem. Finally, Sect. 4.7 evaluates the EMQA on a benchmark dataset.

After training the model, we test it by entering an image as well as a question to the model, and the model can produce several answers as the result. In order to compare our model's output with the target, a new method of evaluation is required. Previous works in VQA are able to output a single answer for a question. However, in our work in the test time, the model outputs multiple answers for the question. Hence, we need a new method to evaluate our approaches. One method of evaluation might be the binary classification accuracy metric, but it is not suitable for our task. The binary classification accuracy takes the true negatives into account while we do not care about the answers that are not being picked correctly in our tests. However, we do care about the precision of the predicted answers and how accurately all true answers are collected. In order to achieve this objective, we apply the F_1 measure [20] for assessing our task. Assuming tp , tn , fp , and fn are consecutively true positive, true negative, false positive and false negative for comparing a single question answers with the ground truth, we can calculate the F_1 measure using the following equations:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

where Precision and Recall are calculated via:

$$\text{Precision} = \frac{tp}{tp + fp},$$

$$\text{Recall} = \frac{tp}{tp + tn}.$$

While the F_1 measure is for a single sample, we define F_1 score as the average of F_1 measures of all samples multiplied by 100.

The models are implemented in Keras framework with the Tensorflow backend. Dropout (between 0.5 and 0.7) and l2 regularization term (with the weight of 0.05) are used to prevent the model from overfitting, while batch normalization and Xavier initialization of the weights help with faster convergence. All of the models are optimized using Adam optimizer, while $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. All images are normalized before training the model by dividing each channel of all pixels by 255. The word embeddings are initiated randomly and learned through the training process.

4.1 Icon question answering (ICQA)

In order to validate our hypothesis, we need a dataset which has images just detailed enough to answer the question. As we searched through the available datasets for the task of VQA, we did not find any dataset with the mentioned properties. So we decided to create our own dataset. We collect

Table 2 Question templates used in ICQA

	Question	Repeats (per image)
1	What is the shape?	1
2	What color is the shape?	1
3	What color is the background?	1
4	What color is the {shape}?	100
5	What is the {color} shape?	21

about 100 different 30×30 resolution iconic shapes from the Internet and also define 21 different colors for the shape and its background. Due to our multiple approaches, we create multiple sets of data and reference to them on desired section. The first set is called *A*. For each sample of this set, we generate an image using a random shape (including no shape), a random shape color and a random background color. It is worth mentioning that the colors of the background and the shape do not match. We place the colored shape randomly over a 64×64 resolution background. Some images of the training set are shown in Fig. 4. Next step is to generate a couple of questions associated with the images. For this task, we design some template questions about the type, color and existence of the shape and also the color of the background. Table 2 shows the questions used in our dataset along with the number of its repeats where curly braces mean that in this case we should iterate over all of the possible values of what is inside it. For example, {shape} means to iterate over all possible values of the shapes which are about 100 different shapes in our dataset. Therefore, there are 124 different questions asked per image. Set *A* contains 260,840 questions asked from 42,021 images.

Since some answers are repeated more than the others, we have to balance our dataset based on the answers to make sure that the corresponding model will be trained on all the answers equally. So we balance the answers based on the question type in such a way that the number of the answers to the color questions should be matched. The same procedure goes for the shape questions.

The next two sets called *B* and *C* have a different scenario. We choose the size of 512×512 for both sets of images. The background of the images is randomly selected and contains one or two colors that split the image vertically or horizontally. After filling the background, we collect a random (between 7 and 15) number of shapes from our shape pool with random colors and place them randomly in our image. Some samples from these sets are demonstrated in Fig. 5. The question templates are the same as set *A*. Set *B* contains 226,406 questions asked from 42,300 large images, while set *C* contains 5408 questions asked from 1000 images.

It is obvious that our dataset is matched to the criteria discussed by Goyal [6], making it a good candidate for future

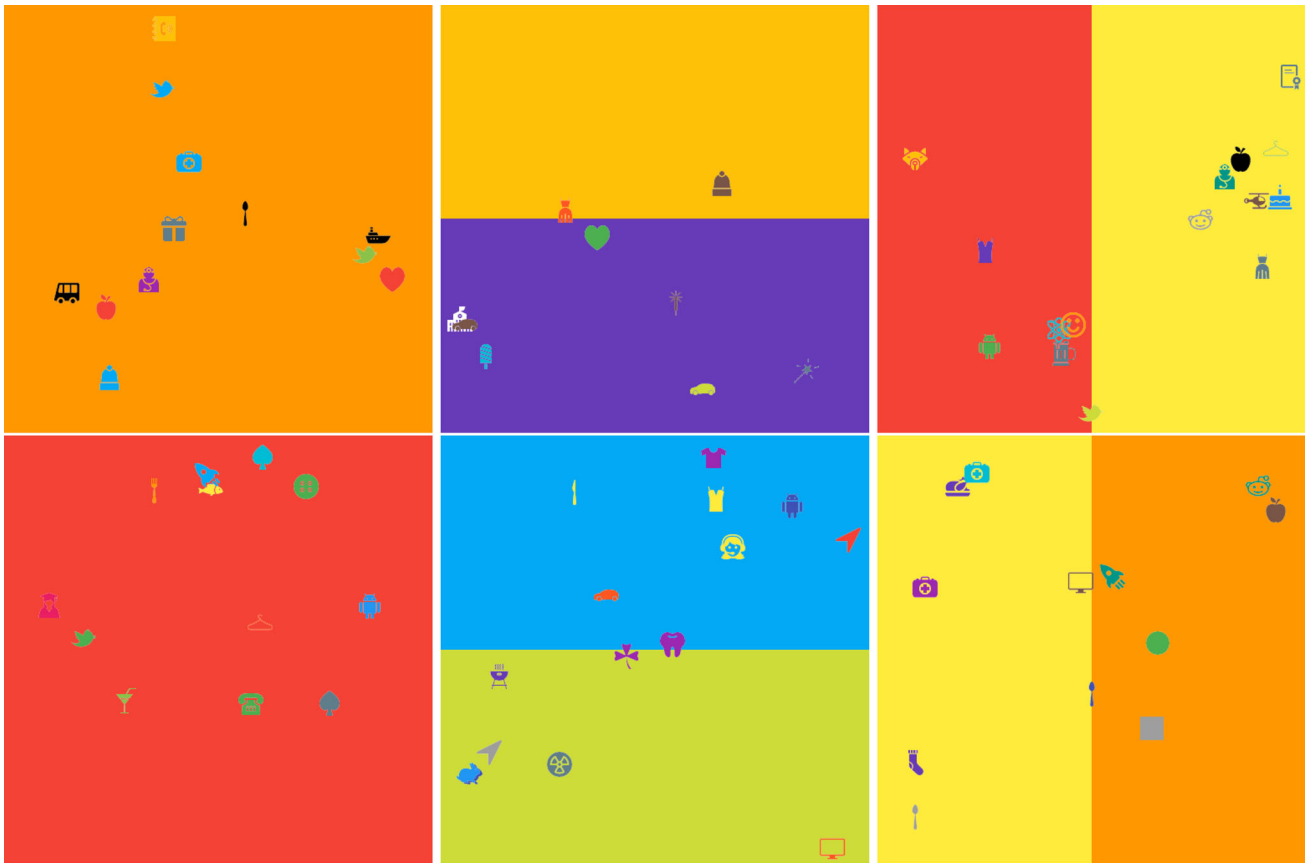


Fig. 5 Six samples of the test images from the ICQA dataset

VQA research. So we will make the dataset available to the public.

4.2 Glimpse training approach evaluation

We train our model along with four different non-attention models on the small images (set *A*) which contain a single answer. So during the training time, the input is a glimpse with a question and the output is the answer to that question, while in the test time we try the large images with the corresponding questions. The output of the model in both training and test time is a vector with the size of 123 (100 classes the shapes, 21 for the colors, one class for no shape and one class for no color). The image embedding part of all models is a couple of trainable convolutional layers which are the same in all models. As we use set *A* for the training, all models are able to get fairly good results on the training set. In the test time, we use set *C* and pass the large images with the questions to our model and use sliding window approach with the window size of 64×64 and the stride of (16, 16) for the other models because they are not able to accept scalable input images. Then, the results are aggregated using the *SoftmaxUnion*. After aggregating, we use the F_1 measure

Table 3 The accuracy on the training and the validation set along with F_1 score and floating point operations on the test set for each candidate model

Model	F_1 score	Test time flops (million operations)
Vis2LSTM	62.25	219
AYN	59.28	330
DPP	64.73	493
BowIMG	46.24	232
EMQA	62.91	19

to get a method of evaluation. Meanwhile, we calculate the amount of operation used in a single feedforward process of each model for a large image and a question. It can be observed in Table 3 that although our model achieves fairly good results in F_1 score, the number of floating point operations is what distinguishes our model from the other methods. Bold items represent the highest performance. Our model is able to reduce the amount of floating point operation about 16 times compared to the average of computational costs of the other methods while maintaining proper accuracy.

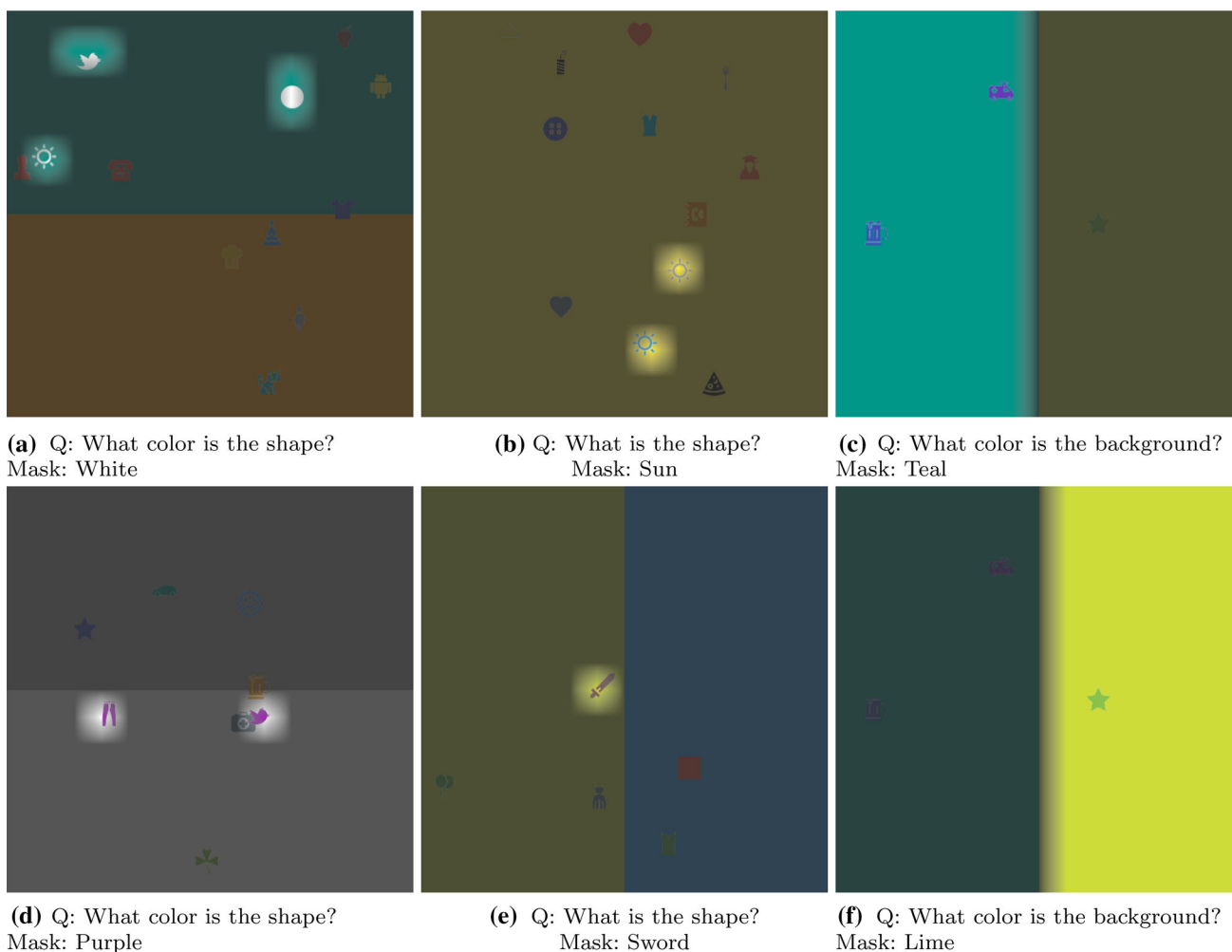


Fig. 6 Samples of the masked test images

4.3 Visualization of EMQA regional output

In this section, we want to make sure that the model is working as expected on the large test images and the aggregated answers are not random. We try to create a mask based on an answer for a single question on some large images. As we mentioned before, our model’s output is a three-dimensional tensor. In order to extract the mask for an answer from the output tensor, we filter out the output tensor based on the answers class to get a spatial matrix. The spatial matrix expresses the probabilities of the answer in different regions of the image. For output tensor T and answer class c , the mask can be noted as $Mask(T, c) = T[:, :, c]$. In order to apply the mask to the input image, we should scale up the spatial matrix to match the size of the input image using an algorithm like bilinear interpolation or proximal interpolation. The result of scaling up is a mask matrix with the size equal to the input images size. Each element of the matrix shows the contribution of a pixel to produce the answer. We apply the mask to the image

Table 4 The results of modality bias test

Input	F1 score
No image	15.32
No question	15.17
No image & no question	2.23

to track down the regions of the input image which have raised the answer. This test will help us visually to ensure that the correct regions have raised the answer. Some of the masked images are shown in Fig. 6. According to our results, we can say that our model extracts the answers from the valid regions of the large images.

4.4 EMQA modality bias evaluation

In this experiment, we design several scenarios to explore whether our model is biased toward either the question or

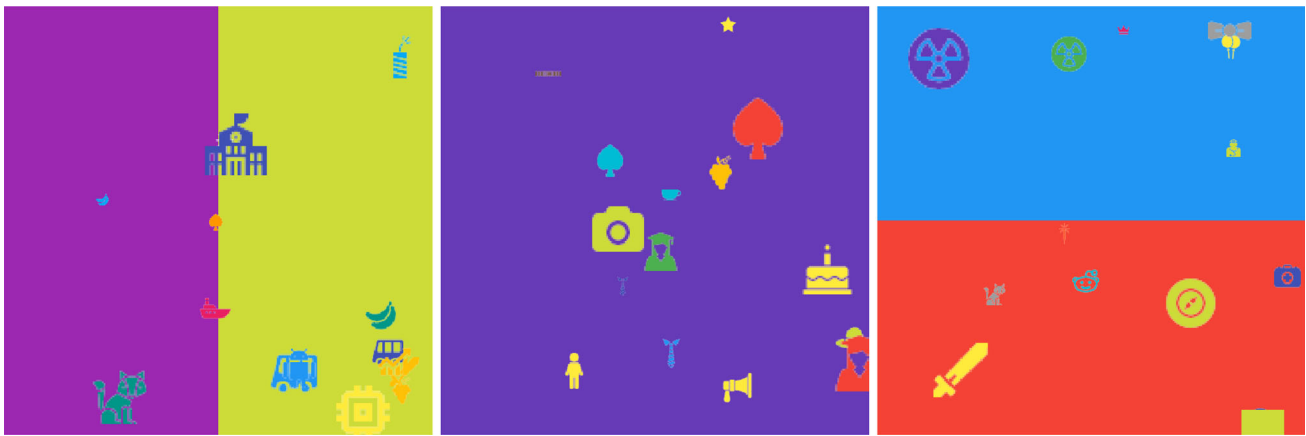


Fig. 7 Three samples of the images of the set D

Table 5 Multi-scale input image evaluation results

Number of input scales (k)	F_1 score
1 scale	40.74
3 scales	43.13
5 scales	44.15
7 scales	45.63

the image. In the first case, we pass each of the questions of the set C with a black image (an image with all pixels set to zero) into our model. In the second case, each of the images of the set C along with an empty question (a vector filled with zero) is fed into the model, and finally in the last case, a black image, as well as an empty question for every sample of the set C , is considered to be entered into the model. The results of these tests are reported in Table 4. From this table, we can understand that neither our model nor our dataset is biased toward any of the modalities and both image and question are required to make a reliable predictions.

4.5 EMQA multi-scale input image evaluation

As we mentioned before, we can pass the multiple scales of a single image to our model and process them with the question in order to perform better in the cases that the object is too small or too large for our training windows size. To evaluate this, we create a new set of images. The set is much like set C except that the iconic shapes are randomly scaled between 0.5 and 2.5 and then are placed onto the background. Hereafter, we call it set D . Some of the images of this set are illustrated in Fig. 7. We train our model using the set A , and in the test time, the images of set D in k different scales are fed into the model where k equals 1, 3, 5 or 7. From the results stated in Table 5, we can infer that changing the scale of the iconic shapes can cause a noticeable amount of loss in F_1 score. However, the

Table 6 F_1 score along with test time floating point operations for each candidate model while training with *sigmoid* activation in the last layer instead of *softmax*

Model	F_1 score	Test time flops (million operations)
MA-Vis2LSTM	20.68	126
MA-AYN	20.08	237
MA-DPP	34.29	458
MA-BowIMG	18.77	228

model can perform better by considering multiple scales of the input image. We also did this test for DPP, as it was the best performer in our previous tests, and noticed that multi-scale DPP follows the same pattern as multi-scale EMQA with slightly higher accuracy. For instance, in the seven scales, the amount of operations used by DPP was 27 times more than EMQA. All of this extra cost is coming for less than 0.5% increased accuracy.

4.6 Multi-class classification approach evaluation

In the single-answer VQA, the output of the models is a *softmax* activated vector that turns the VQA into a classification task in which the class with maximum probability is chosen as the answer. A trivial solution is to use the *sigmoid* nonlinearity instead of *softmax* function as the activation of the last layers turning the multi-answer problem into a multi-class classification task. Using this change, we can make older models feasible for the task of multi-answer VQA without any modified training set. Therefore, we add “MA” (stands for multi-answer) before the name of each model in Table 6 to distinguish them from the original methods. We train every multi-answer model on set B and measure its F_1 score on set C . The results are stated in Table 6. It can be observed that MA-DPP achieves the best result on

Table 7 EMQA evaluation on DAQUAR reduced and DAQUAR full datasets

Model	DAQUAR reduced accuracy (%)	DAQUAR full accuracy (%)
Vis2LSTM	35.77	21.84
AYN	34.66	21.04
DPP	44.5	28.87
BowIMG	26.37	16.17
EMQA	40.19	25.78

the ICQA dataset, while both Tables 3 and 6 determine that glimpse training approach performs significantly better in F_1 score test than multi-class classification approach. This happens because of the omission of unnecessary detail in the image.

4.7 EMQA evaluation on the other datasets

In order to prove that EMQA is capable of learning real-world images, we test it on one of the most popular datasets. We choose to test our model on the DAQUAR [16] dataset as it is a benchmark for the task of VQA. The name stands for DATaset for QUestion Answering on Real-world images. It includes 1449 image for both train and test sets. Questions in DQAUAR are generated by both machine and human, containing the total number of 12,468 questions. A reduced version of DAQUAR is also available which contains 3786 training and 279 test questions. In order to test our model, we remove two of 1×1 convolution layers since DAQUAR is not as complex as our own dataset. The rest of the network architecture stays the same. It is worth mentioning that DAQUAR is a single-answer dataset; hence, we do not apply any glimpse training or multi-answer approach as we just want to test EMQA's capability to learn the real-world images on a benchmark dataset. Our model results can be compared with other non-attention models for both DAQUAR reduced and DAQUAR full datasets in Table 7. Bold items represent the highest performance. Although our model does not outperform the state of the art, its performance is good enough to make sure that it can learn real-world images as well.

5 Limitations and future works

It can be observed that the glimpse training approach learns from fewer data, generalizes more accurately, and demands less computational power compared to the multi-class classification approach. However, there are some limitations that will be discussed along with some potential solutions in the subsequent subsections.

5.1 Limitations

As we mentioned before, the glimpse training approach can only answer object-oriented questions. These types of questions ask about an object or its properties. The questions that ask about the location or count cannot be answered because the glimpse training approach answers questions regarding a local part of an image and these types of questions require a global view of the image. Moreover, the model for answering some questions requires the abstract meaning of the scene. These types of questions are not resolvable by the glimpse training approach.

5.2 Future works

As mentioned in the previous subsection, all of the glimpse training approach limitations come from the fact that the questions are answered locally and the model cannot take advantage of using the global view. Perhaps, a hybrid model that utilizes both multi-class classification (global view) and glimpse approaches (local view) together can be a solution to this problem. The counting problem could be addressed using a binary classifier that classifies the existence of the object at each region of the image as well as a new aggregation of the binary results. Moreover, adapting more complex techniques such as attention-based models [14,26,28–30] to the glimpse training approach can be a beneficial research to increase the accuracy.

6 Conclusion

In this paper, we established a new problem in the field of VQA which was considering multiple answers for a question in an image. We proposed a new training approach called glimpse training. After the glimpse training, the sliding window approach should be used during the test time to answer multi-answer questions. The glimpse training approach increased the F_1 score significantly compared to the multi-class classification approach. Because of the high computational cost of the sliding window approach, we introduced a new model to serve our approach in a more computationally efficient manner. We also created a new dataset designed from the ground up for the purpose of the glimpse training approach. We assessed our model using this dataset, and the results demonstrated that our model is much more computationally efficient than the other methods so that it can reduce the number of floating point operations by 94% making it a feasible choice for the real-time applications. We also tested our model on a popular VQA dataset to confirm that our model is capable of learning the real-world images.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. (2014). arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078)
2. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *Computer Vision-ECCV 2018*, pp. 472–488. Springer, Cham (2018)
3. Dong, X., Shen, J., Wang, W., Liu, Y., Shao, L., Porikli, F.: Hyperparameter optimization for tracking with continuous deep q-learning. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp. 518–527 (2018). <https://doi.org/10.1109/CVPR.2018.00061>
4. Dong, X., Shen, J., Wu, D., Guo, K., Jin, X., Porikli, F.: Quadruplet network with one-shot learning for fast visual object tracking. *IEEE Trans. Image Process.* **28**(7), 3516–3527 (2019). <https://doi.org/10.1109/TIP.2019.2898567>
5. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. (2016). arXiv preprint [arXiv:1606.01847](https://arxiv.org/abs/1606.01847)
6. 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. (2016). arXiv preprint [arXiv:1612.00837](https://arxiv.org/abs/1612.00837)
7. Gupta, A.K.: Survey of visual question answering: datasets and techniques. (2017). CoRR [arXiv:1705.03865](https://arxiv.org/abs/1705.03865)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural comput.* **9**(8), 1735–1780 (1997)
9. Kabbai, L., Abdellaoui, M., Douik, A.: Image classification by combining local and global features. *Vis. Comput.* **35**(5), 679–693 (2019). <https://doi.org/10.1007/s00371-018-1503-0>
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc. (2012). <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
11. Lasecki, W.S., Thiha, P., Zhong, Y., Brady, E., Bigham, J.P.: Answering visual questions with conversational crowd assistants. In: *Proceedings of the 15th international ACM SIGACCESS conference on computers and accessibility, ASSETS '13*, pp. 18:1–18:8. ACM, New York (2013). <https://doi.org/10.1145/2513383.2517033>
12. LeCun, Y., Haffner, P., Bottou, L., Bengio, Y.: Object recognition with gradient-based learning. *Shape, Contour and Grouping in Computer Vision*, pp. 319–345. Springer, Berlin (1999)
13. Liu, X., Xu, Q., Wang, N.: A survey on deep neural network-based image captioning. *Vis. Comput.* **35**(3), 445–470 (2019). <https://doi.org/10.1007/s00371-018-1566-y>
14. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: *Advances in Neural Information Processing Systems*, pp. 289–297 (2016)
15. Ma, L., Lu, Z., Li, H.: Learning to answer questions from image using convolutional neural network. In: *AAAI*, vol. 3, p. 16 (2016)
16. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems 27*, pp. 1682–1690. Curran Associates, Inc. (2014). <http://papers.nips.cc/paper/5411-a-multi-world-approach-to-question-answering-about-real-world-scenes-based-on-uncertain-input.pdf>
17. Malinowski, M., Rohrbach, M., Fritz, M.: Ask your neurons: a neural-based approach to answering questions about images. In: *Proceedings of the IEEE international conference on computer vision*, pp. 1–9 (2015)
18. Noh, H., Hongsuck Seo, P., Han, B.: Image question answering using convolutional neural network with dynamic parameter prediction. In: *The IEEE conference on computer vision and pattern recognition (CVPR)* (2016)
19. Ren, M., Kiros, R., Zemel, R.: Exploring models and data for image question answering. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 28*, pp. 2953–2961. Curran Associates, Inc. (2015). <http://papers.nips.cc/paper/5640-exploring-models-and-data-for-image-question-answering.pdf>
20. Rothschild, A.S., Hripcsak, G.: Agreement, the F-measure, and reliability in information retrieval. *J. Am. Med. Inf. Assoc.* **12**(3), 296–298 (2005). <https://doi.org/10.1197/jamia.M1733>
21. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. (2013). CoRR [arXiv:1312.6229](https://arxiv.org/abs/1312.6229)
22. Shih, K.J., Singh, S., Hoiem, D.: Where to look: focus regions for visual question answering. In: *The IEEE conference on computer vision and pattern recognition (CVPR)* (2016)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. (2014). CoRR [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
24. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. (2014). CoRR [arXiv:1409.4842](https://arxiv.org/abs/1409.4842)
25. Tripathi, G., Singh, K., Vishwakarma, D.K.: Convolutional neural networks for crowd behaviour analysis: a survey. *Vis. Comput.* **35**(5), 753–776 (2019). <https://doi.org/10.1007/s00371-018-1499-5>
26. Wang, W., Shen, J.: Deep visual attention prediction. *IEEE Trans. Image Process.* **27**(5), 2368–2378 (2018). <https://doi.org/10.1109/TIP.2017.2787612>
27. Wang, W., Shen, J., Ling, H.: A deep network solution for attention and aesthetics aware photo cropping. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(7), 1531–1544 (2019). <https://doi.org/10.1109/TPAMI.2018.2840724>
28. Wang, W., Shen, J., Shao, L.: Video salient object detection via fully convolutional networks. *IEEE Trans. Image Process.* **27**(1), 38–49 (2018). <https://doi.org/10.1109/TIP.2017.2754941>
29. Wang, W., Shen, J., Xie, J., Cheng, M., Ling, H., Borji, A.: Revisiting video saliency prediction in the deep learning era. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019). <https://doi.org/10.1109/TPAMI.2019.2924417>
30. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 21–29 (2016)
31. Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., Fergus, R.: Simple baseline for visual question answering. (2015). arXiv preprint [arXiv:1512.02167](https://arxiv.org/abs/1512.02167)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Sayedshayan Hashemi Hosseini was born in Esfahan. He received his B.Sc. degree in Computer Engineering from Islamic Azad University, Najafabad Branch, Iran, in 2016. He also received her M.Sc. degree in Computer Engineering from Isfahan University of Technology, Isfahan, Iran, in 2019. His research interests include deep learning and computer vision.



Mehran Safayani received his B.S. degree in Computer Engineering from Isfahan University, Isfahan, Iran, in 2002. Then, he received the M.Sc. and Ph.D. degrees from Sharif University of Technology, Tehran, Iran, in Computer Architecture and Artificial Intelligence in 2006 and 2011, respectively. Since 2012, he is an assistant professor of Electrical and Computer Engineering at Isfahan University of Technology. His research interests include machine learning, neural networks and deep learning,

pattern recognition and soft computing.



Abdolreza Mirzaei was born in Isfahan, Iran. He received the B.Sc. (first-class honors) degree in Computer Engineering from Isfahan University, in 2001, the M.Sc. degree in Artificial Intelligence from Iran University of Science and Technology, Tehran, Iran, in 2003, and the Ph.D. degree in Artificial Intelligence from Amirkabir University of Technology, Tehran, in 2009, respectively. He is currently in the Department of Electrical and Computer Engineering, Isfahan University of Technology. His research

interests include statistical and structural classification methods, digital image processing, computer vision, multiple classifier systems and learning methods.