Journal of Big Data

RESEARCH Open Access

Image captioning model using attention and object features to mimic human image understanding

Muhammad Abdelhadie Al-Malla[1*], Assef Jafar[1] and Nada Ghneim[2]

*Correspondence: abdelhadie.almalla@hiast.edu.sy
[1] Present Address: Higher Institute for Applied Sciences and Technology, Syria, Damascus
Full list of author information is available at the end of the article

Abstract

Image captioning spans the fields of computer vision and natural language processing. The image captioning task generalizes object detection where the descriptions are a single word. Recently, most research on image captioning has focused on deep learning techniques, especially Encoder-Decoder models with Convolutional Neural Network (CNN) feature extraction. However, few works have tried using object detection features to increase the quality of the generated captions. This paper presents an attention-based, Encoder-Decoder deep architecture that makes use of convolutional features extracted from a CNN model pre-trained on ImageNet (Xception), together with object features extracted from the YOLOv4 model, pre-trained on MS COCO. This paper also introduces a new positional encoding scheme for object features, the "importance factor". Our model was tested on the MS COCO and Flickr30k datasets, and the performance is compared to performance in similar works. Our new feature extraction scheme raises the CIDEr score by 15.04%. The code is available at: https://github.com/abdelhadie-almalla/image_captioning

Keywords: Image captioning, Object features, Convolutional neural network, Deep learning

Introduction

The topic of autonomously producing descriptive sentences for images has stimulated interest in natural language processing and computer vision research in recent years. Image captioning is a key task that necessitates a semantic comprehension of images as well as the capacity to generate accurate and precise description sentences.

In the big data era, images are one of the most available data types on the Internet, and the need for annotating and labeling them increased. Thus, image captioning systems are an example of big data problems as they focus on the volume aspect of big data. For example, the MS COCO dataset contains around 123,000 images (25 GB). This adds the requirement of efficient use of resources and careful design of experiments.

Early approaches used template methods, trying to fill predefined templates of text by features extracted from images [1]. Current systems benefit from the available computing power and use deep learning techniques.

One of the most successful methods for image captioning is implementing an Encoder-Decoder architecture. It encodes images to a high-level representation then decodes this representation using a language generation model, like Long Short-Term Memory (LSTM) [2], Gated Recurrent Unit (GRU) [3] or one of their variants.

The attention mechanism has demonstrated its effectiveness in sequence-to-sequence applications, especially image captioning and machine translation. It increases accuracy by forcing the model to concentrate on the important parts of the input when generating output sequences [4].

To understand an image, many modern deep learning models use existing pre-trained Convolutional Neural Networks (CNNs) to extract matrices of features from the last convolutional layers. This helps to grasp many aspects of the objects and their relationships in the picture and represent the image at a higher level [5].

Recently, some works tried to use object features for image captioning. Among the models used are the YOLOv3 [6], YOLOv4 [7] and YOLO9000 [8], which are known for their speed, accuracy and effectiveness for real-time applications. Object features are usually an array of object tags, where each object tag contains the bounding box information, object class and confidence rate. This work investigates the hypothesis that exploiting such features could increase accuracy in image captioning and that using all object features helps to accurately mimic the human visual understanding of scenes. This paper aims to present a model that makes use of this type of features through a simple architecture and evaluate the results.

Section two of the paper will tackle the related works in the domain. Section three presents our methodology, which includes the proposed model and the pre-processing that was performed on the data. In section four, the experiments design and results are elaborated, and a comparison to previous works is shown. Section five concludes the paper and presents plans for future works.

Related works

In [9], Yin and Ordonez suggested a sequence-to-sequence model in which an LSTM network encodes a series of objects and their positions as an input sequence and an LSTM language model decodes this representation to generate captions. Their model uses the YOLO [8] object detection model to extract object layouts from images (object categories and locations) and increase the accuracy of captions. They also present a variation that uses the VGG [10] image classification model pre-trained on ImageNet [11] to extract visual features. The encoder at each time step takes as input a pair of object category (encoded as a one-hot vector), and the location configuration vector that contains the left-most position, top-most position, width and height of the bounding box corresponding to the object, all normalized. The model is trained with back-propagation, but the error is not propagated to the object detection model. They showed that their model increased in accuracy when combined with CNN and YOLO modules. They did not use all available data from the object features produced by YOLO, such as object dimensions and confidence.

In [12] Vo-Ho et al. developed an image captioning system that extracts object features from YOLO9000 [8] and Faster R-CNN [13]. Each type of features is processed through an attention module to produce local features that represent the part that the model is currently focusing on. The two local feature sets are combined and fed into an LSTM model to generate the probabilities of the words in the vocabulary set at each time step. A beam search strategy is used to process the results, in order to choose the best candidate caption. They used the ResNet [14] CNN to extract the features from images. From a given image as input, they first extracted a list of tags using YOLO9000, then break each tag into words and eliminate redundant ones so the list will contain only unique words. Each word i, including the "null" token, is represented by a one-hot vector of the size of the vocabulary set. After that, they embed each word into a d-dimension space using the word embedding method. They used LSTM units for language generation. They only keep the top 20 tags with the highest probabilities.

In [15], Lanzendörfer et al. proposed a model for Visual Question Answering (VQA) based on iBOWIMG. The model extracts features from Inception V3 [16] as well as object features extracted from the YOLO [8] object detection model, and uses the attention mechanism. The outputs of YOLO are encoded as vectors of size 80×1 in order to give more informative features to the iBOWIMG model, with each column containing the number of detected objects of the given type. Three of these object vectors are produced for detection confidence thresholds of 25%, 50% and 75% and then concatenated with the image features and question features.

In [17], Herdade et al. proposed a spatial attention-based encoder-decoder model that explicitly integrates information about the spatial relationship between detected objects. They employed an object detector to extract appearance and geometry features from all detected objects in the image, then the Object Relation Transformer to generate caption text. They used Faster R-CNN [13] with ResNet-101 [14] as the base CNN for object detection and feature extraction. A Region Proposal Network (RPN) generates bounding boxes for object proposals using intermediate feature maps from the ResNet-101 as inputs. Overlapping bounding boxes with an intersection-over-union (IoU) exceeding a threshold of 0.7 are discarded, using non-maximum suppression. All bounding boxes where the class prediction probability is below a threshold of 0.2 are also discarded. Then, for each object bounding box, they perform mean-pooling over the spatial dimension to build a 2048-dimensional feature vector. These feature vectors are then input to the Transformer model.

In [18] Wang et al. studied end-to-end image captioning with highly interpretable representations obtained from explicit object detection. They performed a detailed review of the effectiveness of a number of object detection-based cues for image captioning. They discovered that frequency counts, object size, and location are all useful and complement the accuracy of the captions produced. They also discovered that certain object categories had a greater effect than others on image captioning.

The work of Sharif et al. [19] suggested to leverage the linguistic relations between objects in an image to boost image captioning quality. They leverage "word embeddings" to capture word semantics and capsulize the semantic relatedness of objects. The proposed model uses linguistically-aware relationship embeddings to capture the spatial and semantic proximity of object pairs. It also uses NASNet to capture the image's global semantics. As a result, true semantic relations that are not apparent in an image's visual content can be learned, allowing the decoder to focus on the most important object relations and visual features, resulting in more semantically-meaningful captions.

Variš et al. [20] investigated the possibility of textual and visual modalities sharing a common embedding space. They presented an approach that takes advantage of object detection labels' textual nature as well as the possible expressiveness of visual object representations built from them. They investigated whether grounding the representations in the captioning system's word embedding space, rather than grounding words or sentences in their associated images, could improve the captioning system's efficiency. Their proposed grounding approaches ensure that the predicted object features and the term embedding space are mutually grounded.

Alkalouti and Masre [21] proposed a model to automate video captioning based on an Encoder-Decoder architecture. They first select the most important frames from the video and remove redundant ones. They used the YOLO model to detect objects in video frames and an LSTM model for language generation.

In [22], Ke et al. investigated the feature extraction performance of 16 popular CNNs on a dataset of chest X-ray images. They did not find a relationship between the performance on ImageNet and the performance on the medical image dataset. However, they found out that the choice of CNN architecture influences performance more than the concrete model within the model family for medical tasks. They also noticed that ImageNet pre-training gives a boost to performance in all architectures, with a lower boost for bigger architectures. They also observed that ImageNet pre-training yields a statistically significant boost in performance across architectures, with a higher boost for smaller architectures.

In [23], Xu et al. proposed a novel Anchor-Captioner method. They started by identifying the significant tokens that should be given more attention and using them as anchors. The relevant texts for each chosen anchor were then grouped to create the associated anchor-centered graph (ACG). Finally, they implemented multi-view caption generation based on various ACGs in order to improve the content diversity of generated captions.

In [24], Chen et al. suggested Verb-specific Semantic Roles (VSR) as a new Controllable Image Captioning (CIC) control signal. VSR is made up of a verb and some semantic roles that reflect a specific activity and the roles of the entities involved in it. They trained a Grounded Semantic Role Labeling (GSRL) model to locate and ground all entities associated with each role given a VSR. Then, to learn human-like descriptive semantic structures, they suggested a Semantic Structure Planner (SSP). Lastly, they used a role-shift captioning model to generate the captions.

In [25], Cornia et al. presented a unique framework for image captioning, which allows both grounding and controllability to generate diverse descriptions. They produced the relevant caption using a recurrent architecture that explicitly predicts textual chunks based on regions and adheres to the control's limitations, given a control signal in the form of a series or a collection of image regions. Experiments are carried out using Flickr30k Entities and COCO Entities, a more advanced version of COCO that includes semi-automated grounding annotations. Their findings showed that the method produces state-of-the-art outcomes in terms of caption quality and diversity for controllable image captioning.

Unlike previous works, our approach takes advantage of all object features available. The experiments section shows the effect of this scheme.

Table 1 A comparison of the used datasets

Dataset	Training split	Validation split	Testing split	Total images
Flickr30k	28k	1k	1k	30k
MS COCO	83k	41k	41k	144k

Research methodology

The experimental method involves extracting object features from the YOLO model and introducing them along with CNN convolutional features to a simple deep learning model that uses the widespread Encoder-Decoder architecture with the attention mechanism. "Results and discussion" section compares the difference in results before and after adding the object features. Although previous research encoded object features as a vector, we add object features in a simple concatenation manner and achieved a good improvement. We also test the impact of sorting the object tags extracted from YOLO according to a metric that we propose here.

Datasets used

We test our method on two datasets used usually for image captioning: MS COCO and Flickr30k. Table 1 contains a brief comparison between them. They are both collected from the Flickr photo sharing website and consist of real-life images, annotated by humans (five annotations per image).

It is worth noting that MS COCO does not publish the labels of the testing set.

Evaluation metrics

We use a set of evaluation metrics that are widely used in the image captioning field. BLEU [26] metrics are commonly used in automated text evaluation and quantify the correspondence between a machine translation output and a human translation; in the case of image captioning, the machine translation output corresponds to the automatically produced caption, and the human translation corresponds to the human description of the image. METEOR [27] is computed using the harmonic mean of unigram precision and recall, with the recall having a higher weight than the precision, as follows:

$$METEOR = [10 * Precision * Recall/(Recall + 9 * Precision)]$$

ROUGE-L [28] uses a Longest Common Subsequence (LCS) score to assess the adequacy and fluency of the produced text, while CIDEr [29] focuses on grammaticality and saliency. SPICE [30] evaluates the semantics of the produced text by creating a "scene graph" for both the original and generated captions, and then only matches the terms if their lemmatized WordNet representations are identical. BLEU, METEOR, and ROUGE have low correlations with human quality tests, while SPICE and CIDEr have a better correlation but are more difficult to optimize.

Model

Our model uses an attention-based Encoder-Decoder architecture. It has two methods of feature extraction for image captioning: an image classification CNN (Xception [31]), and an object detection model (YOLOv4 [7]). The outputs of these models are

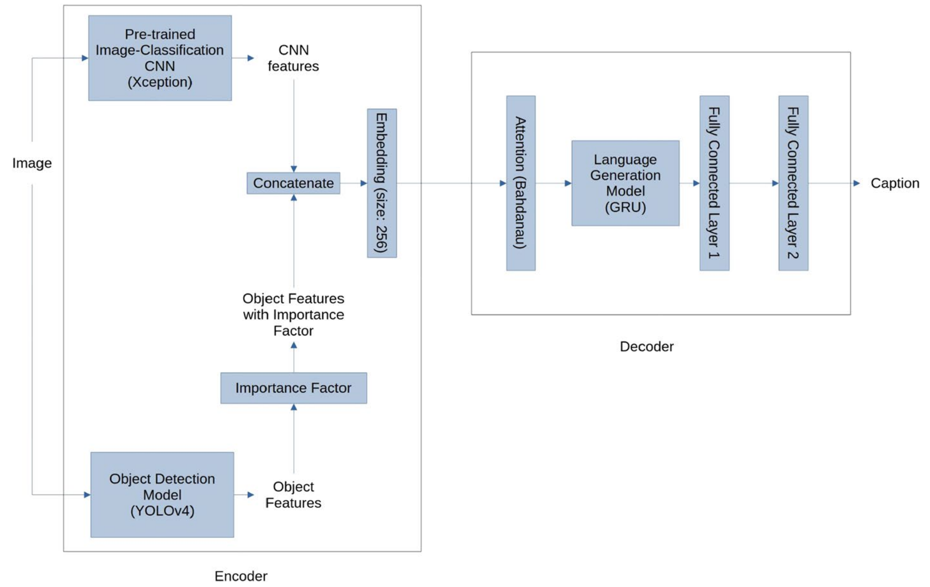


Fig. 1 A block diagram of the proposed model

combined by concatenation to produce a feature matrix that carries more information to the language decoder to predict more accurate descriptions. Unlike others' works that embedded object features before combining them with CNN features, we use raw object layout information directly. Language generation is done using an attention module (Bahdanau attention [32]), a GRU [3] and two fully connected layers. Our model is simple, fast to train and evaluate, and generates captions using attention.

We believe that if humans can benefit from object features (such as the class of object, its position, and size) to better understand an image, a computer model can benefit from this information as well. A scene containing a group of people standing close together, for example, may suggest a meeting, whereas sparse crowds can indicate a public location. Figure 1 depicts our model.

Image encoding

A. Pre-trained image classification CNN In this work, we use the Xception CNN pre-trained on ImageNet [11] to extract spatial features.

Xception [31] (Extreme version of Inception) is inspired by Inception V3 [16], but instead of Inception modules, it has 71 layers with a modified depth-wise separable convolution. It outperforms Inception V3 thanks to better model parameter usage.

We extract features from the last layer before the fully connected layer, following recent works in image captioning. This allows the overall model to gain insight about the objects in the image and the relationships between them instead of just focusing on the image class.

In a previous work, different feature extraction CNN models have been compared for image captioning applications. The results showed that Xception was among the most robust in extracting features and for this it was chosen as the feature extraction model in this study. The output of this stage is of shape ($10 \times 10 \times 2048$), which was squashed to (100×2048) for ease of matrix handling.

B. Object detection model Our method uses the YOLOv4 [7] model because of its speed and good accuracy, which make it suitable for big data and real-time applications. The extracted features are a list of object features, with every object feature containing the X coordinate, Y coordinate, width, height, confidence rate (from 0 to 1 inclusive), class number and a novel optional "importance factor".

Following human intuition, foreground objects are normally larger and more important when describing an image, and background objects are normally smaller and less important. Furthermore, it makes sense to use more accurate pieces of information than to use less accurate ones. Hence, our importance factor tries to balance the importance of the foreground large objects and objects with high confidence rates. The formula to calculate it for a single object is as follows:

$$\text{Importance Factor} = \text{Confidence Rate} \times \text{Object Width} \times \text{Object Height}$$

The importance factor gives a higher score to foreground large objects over background small ones, and higher score to objects with a high confidence over objects with less confidence.

After extracting object features, the importance factor is calculated for each object and concatenated to its tag. Then, all objects in the list are sorted according to this importance factor using the quick sort algorithm. Unlike previous works, our method makes use of all of the image's object information. Because of the size restriction in the output of the CNN, we use up to 292 objects, each with seven attributes (including the importance factor), which is usually enough to represent important objects in an image.

The list of features is flattened into a 1D array, of length less than 2048. It is then padded with zeros to length 2048 to be compatible with the output of the CNN module. The output of this stage is an array (1×2048).

As for calculating the confidence score, YOLO divides an image into a grid. B bounding boxes and confidence scores for these boxes are predicted in each of these grid cells. The confidence score indicates how confident the model is that the box includes an object, as well as how accurate the model believes the box that predicted is. The object detection algorithm is evaluated using Intersection over Union (IoU) between the predicted box and the ground truth. It analyzes how similar the predicted box is to the ground truth by calculating the overlap between the ground truth and the predicted bounding box. A cell's confidence score should be zero if no object exists in there. The formula for calculating the confidence score is:

$$\mathrm{C} = \mathrm{Pr}\left(\text{object}\right) * \mathrm{IoU}$$

C. Concatenation and embedding In order to take advantage of the image classification features and the object detection features, we add this concatenation step, where we attach the output of the YOLOv4 subsystem as the last row in the output of stage 1. The output of this stage is of shape (101×2048).

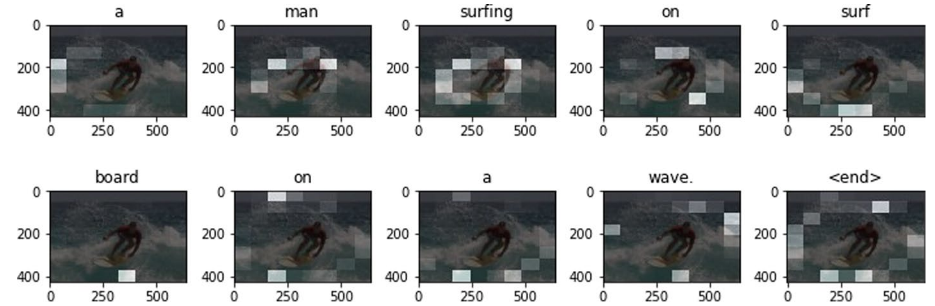


Fig. 2 Example of attending over an image to generate a caption. Image source: https://www.tensorflow.org/tutorials/text/image_captioning

The embedding is done using one fully connected layer of length 256. This stage ensures a consistent size of the features and maps the feature space to a smaller space appropriate for the language decoder.

D. Attention Our method uses the Bahdanau soft attention system [32]. This deterministic attention mechanism makes the model as a whole smooth and differentiable.

The term “attention” refers to a strategy that simulates cognitive attention. The effect highlights the most important parts of the input data while fading the rest. The concept is that the network should dedicate greater computer resources to that small but critical portion of the data. Which component of the data is more relevant than others is determined by the context and is learned by gradient descent using training data. Natural language processing and computer vision use attention in a number of machine learning tasks.

The attention mechanism was created to increase the performance of the encoder-decoder architecture for machine translation. And as image captioning can be viewed as a specific case of machine translation, attention proved useful when analyzing images as well. The attention mechanism was intended to allow the decoder to use the most relevant parts of the input sequence in a flexible manner by combining all of the encoded input vectors into a weighted combination, with the most relevant vectors receiving the highest weights.

Attention follows the human intuition of focusing on different parts of an image when describing it. Using object detection features also follows the intuition that knowing about object classes and positions help to grasp more about the image than mere convolutional features. When attention is employed to both feature types, the system will focus on different features of both object classes and positions in the same image. Figure 2 depicts using attention for image captioning.

Language decoder

For decoding, a GRU [3] is used to exploit its speed and low memory usage. It produces a caption by generating one word at every time step, conditioned on a context vector, the previous hidden state, and the previously generated words. The model is trained using the backpropagation algorithm deterministically.

The GRU is followed by two fully connected layers. The first one is of length 512, and the second one is of the size of the vocabulary to produce output text.

The training process for the decoder is as follows:

1. The features are extracted then passed through the encoder.
2. The decoder receives the encoder output, hidden state (initialized to 0), and decoder input (which is the start token).
3. The decoder returns the predictions as well as the hidden state of the decoder.
4. The hidden state of the decoder is then passed back into the model, and the loss is calculated using the predictions.
5. To determine the next decoder input, “teacher forcing” is employed, which is a technique that passes the target word as the next input to the decoder.

Pre-processing

This section presents the pre-processing algorithm that was performed on the data:

1. Sort the dataset at random into image-caption pairs. This helps the training process to converge fast and prevents any bias during the training. Therefore, preventing the model from learning the order of training.
2. Read and decode the images.
3. Resize the images to the CNN requirements: whatever the size of the image is, it is resized to 299 × 299 as required by the Xception CNN model.
4. Tokenization of the text. Tokenization breaks the raw text into words, that are separated by punctuations, special characters, or white spaces. The separators are discarded.
5. Count the tokens, sort them by frequency and choose the top 15,000 most common words as the system’s vocabulary. This avoids over-fitting by eliminating terms that are not likely to be useful.
6. Generate word-to-index and index-to-word structures. They are then used to translate token sequences into word identifier sequences.
7. Padding. As sentences can be different in length, we need to have the inputs with the same size, this is where the padding is necessary. Here, identifier sequences are padded at the end with null tokens to ensure that they are all the of same length.

Results and discussion

Our code is written in the Python programming language using TensorFlow[1]. library The CNN implementation and trained model were imported from Keras[2]. library, and a YOLOv4 model pre-trained on MS COCO was imported from the yolov4 library[3]. This work uses the MS COCO evaluation tool to calculate scores[4].

[1] Available at https://www.tensorflow.org.

[2] Available at https://keras.io/api/applications.

[3] Available at https://pypi.org/project/yolov4.

[4] Available at https://github.com/tylin/coco-caption.

Table 2 Results of adding object features to the baseline model on MS COCO Karpathy split

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L	SPICE
Baseline model	0.463	0.273	0.156	0.087	0.157	0.339	0.345	0.102
Ours (with YOLO bounding boxes, without the importance factor)	0.486	0.293	0.173	0.099	0.164	0.390	0.358	0.108
Ours (with YOLO bounding boxes and the importance factor)	0.492	0.296	0.174	0.101	0.163	0.390	0.358	0.108
Increase due to the importance factor (%)	1.23	1.02	0.57	2.02	− 0.99	0	0	0
Increase over the baseline model (%)	6.26	8.42	11.53	16.09	3.82	15.04	3.76	5.88

Table 3 Comparison with the results of Yin and Ordonez [9] on MS COCO Karpathy testing split

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L	SPICE
Yin and Ordonez [9] baseline model	NA	NA	NA	0.21	0.215	0.759	0.464	NA
Yin and Ordonez [9] results with object features	NA	NA	NA	0.253	0.238	0.922	0.507	NA
Yin and Ordonez [9] increase (%)	NA	NA	NA	**20.47**	**10.69**	**21.47**	**9.26**	NA
Our increase (%)	**6.26**	**8.42**	**11.53**	16.09	3.82	15.04	3.76	**5.88**

Values in boldface represent the higher increase in the column

Tests are conducted on two widely used datasets for image caption generation: MS COCO and Flickr30k. Every image has five reference captions in these two datasets, which contain 123,000 and 31,000 images, respectively. For MS COCO, 5000 images are reserved for validation and 5000 images are reserved for checking according to Karpathy's split [33]. In the case of Flicker30k dataset, 29,000 images are used for preparation, 1000 for validation, and 1000 for testing. The model was trained for 20 epochs and used Sparse Categorical Cross Entropy as the loss function. For the optimizer, Adam optimizer was employed.

Table 2 presents the results of the proposed model on MS COCO Karpathy split and compares them to the results of the baseline model with features only from Xception. It can be noticed how well the evaluation scores increase after adding object features to the model, especially the CIDEr score, which increased by 15.04%. This reflects good improvement in correlation with human judgment when using full object features, and boosted grammatical integrity and saliency. It appears that the importance factor increases the BLEU metrics and decreases METEOR slightly, whereas the other metric values stay the same. Unlike the findings of Herdade et al. [17], our artificial positional encoding scheme did not decrease the CIDEr score. They tested multiple artificial positional encoding schemes and compared them to their geometric attention mechanism.

To show the effectiveness of our method, we compare our increase in results (with the importance factor) to the increase in results of Yin and Ordonez [9] on MS COCO Karpathy split in Table 3. They also measured the effects of incorporating object features on image captioning results. Their object feature extraction method extracts object layouts from the YOLO9000 model, encodes them through an LSTM

Table 4 A comparison with the results of Sharif et al. [19] on Flickr30k testing split

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L	SPICE
Sharif et al.'s baseline model	0.4368	NA	NA	NA	0.1297	0.2517	0.2997	0.0700
Sharif et al.'s suggested model	0.4462	NA	NA	NA	0.1350	0.2835	0.3116	0.0741
Our baseline model	0.3990	0.2200	0.1170	0.0620	0.1230	0.1480	0.2930	0.0740
Our model (with the importance factor)	0.3980	0.2210	0.1160	0.0610	0.1290	0.1500	0.2980	0.0740
Sharif et al.'s increase (%)	**2.15**	NA	NA	NA	4.08	**12.63**	**3.97**	**5.85**
Our increase (%)	− 0.25	**0.45**	**− 0.86**	**− 1.63**	**4.87**	1.35	1.7	0

Values in boldface represent the higher increase in the column

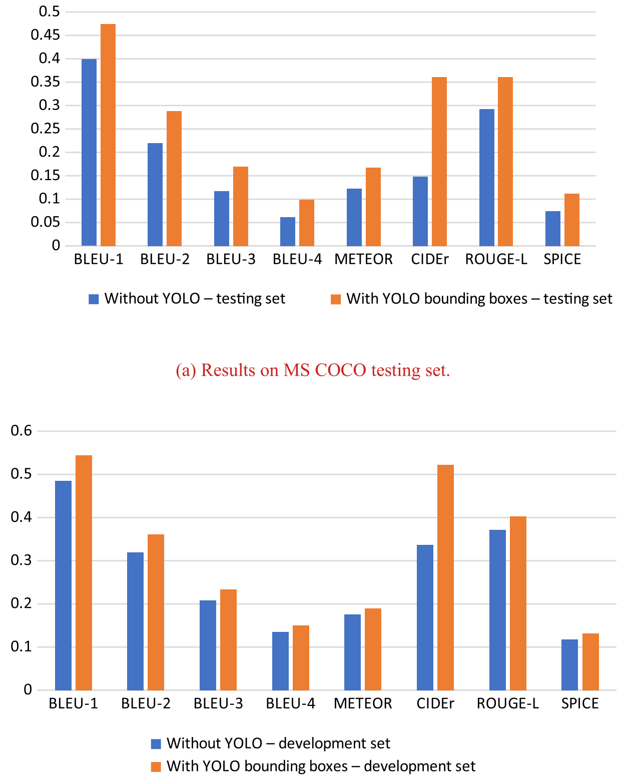
unit, extracts CNN features from the VGG16 CNN, encodes CNN features through another LSTM unit, and sums up the resulting two encoded vectors. We, on the other hand, use raw object features and concatenate them to the CNN features before embedding. We performed the experiments on MS COCO Karpathy [33] testing split (5000 testing images). The highest score difference in each column is written in boldface. Their model gives equal importance to CNN features and object features by passing each type of features to a separate encoding LSTM and then adding them up. Their baseline model has higher accuracy than ours, which may justify the difference between our scores and theirs. They did not report the BLEU-1, BLEU-2, BLEU-3 or SPICE score.

We notice in Table 3 that our results are somewhat comparable to those of Yin and Ordonez [9]. We report all eight standard evaluation scores. The introduction of this type of feature extraction improves all evaluation scores over our baseline model. The increase in the SPICE score (5.88%) reflects increased semantic correlation when using object features, an expected consequence of feeding object tags into the model. SPICE is one of those metrics that are harder to optimize. The score difference between our model and theirs may be related to the feature combination and encoding method. They encode each feature type in a vector, and then add the two vectors, while our model concatenates the two feature sets directly.

We also compare our work with the work of Sharif et al. [19], who tried to benefit from linguistic relations between objects in an image, and we present a comparison between our model and theirs on the Flickr30k dataset [34] in Table 4. We can notice in Table 4 that our method also yields improvement on Flickr30k, with the bigger improvement being in the METEOR score. Sharif et al. benefited from linguistic information in addition to object detection features.

Figure 3 displays a comparison between the baseline model and the model enhanced with object features, on MS COCO Karpathy split [33] validation testing sets. We see a clear increase in the results on all evaluation metrics on both sets, which indicates low generalization error and proves our hypothesis that enhancing the vision model with object detection features improves accuracy.

In order to qualitatively compare the textual outputs of the approach, we present in Fig. 4 a qualitative comparison between the results with object features and without them. We notice that the difference is remarkable, and the addition of the object features makes the sentences more salient grammatically, and with less object mistakes. In (a) for example, a skier was identified instead of just the skiing boots. In (b), the model before



(a) Results on MS COCO testing set.

(b) Results on MS COCO development set.

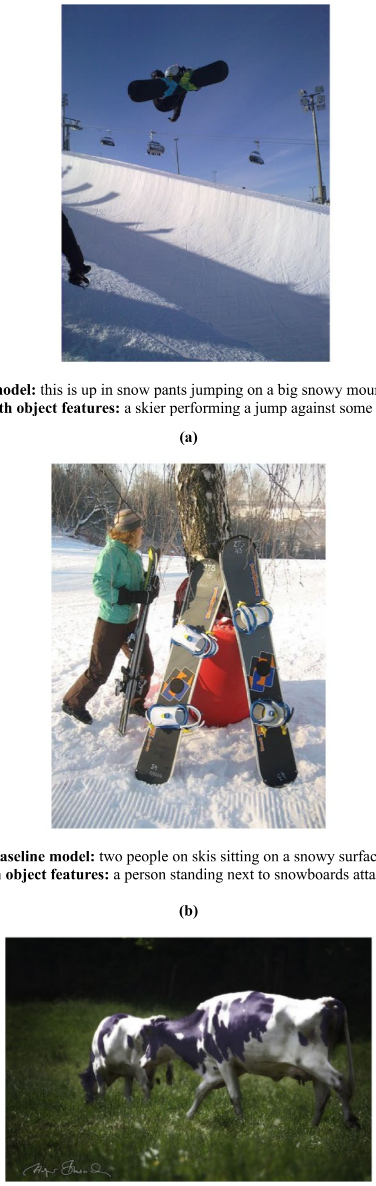
Fig. 3 Comparison between our baseline model (without object features) and our proposed model. **a** Results on MS COCO testing set. **b** Results on MS COCO development set

incorporating object features had mixed up people and snow boards. In (c), the two cows were correctly identified after adding object features. In (d), The model without object features falsely identified a man in the picture. In (e), the model could not identify the third bear without object features. In (f), object features helped to identify a group of people instead of only two women.

Conclusions

In this paper, we presented an attention-based Encoder-Decoder image captioning model that uses two methods of feature extraction, an image classification CNN (Xception) and an object detection module (YOLOv4), and proved the effectiveness of this scheme. We introduced the importance factor, which prioritizes foreground large objects over background small ones, and favors objects with high confidence over those with low confidence and demonstrated its effect on increasing scores. We showed how our method improved the scores and compared it to previous works in the score increase, especially the CIDEr metric which increased by 15.04%, reflecting improved grammatical saliency.

Unlike previous works, our work suggested to benefit from all object detection features extracted from YOLO and showed the effect of sorting the extracted object tags. This can be further improved by better methods for combining object detection features



Baseline model: this is up in snow pants jumping on a big snowy mountain at night.
With object features: a skier performing a jump against some snow.

(a)

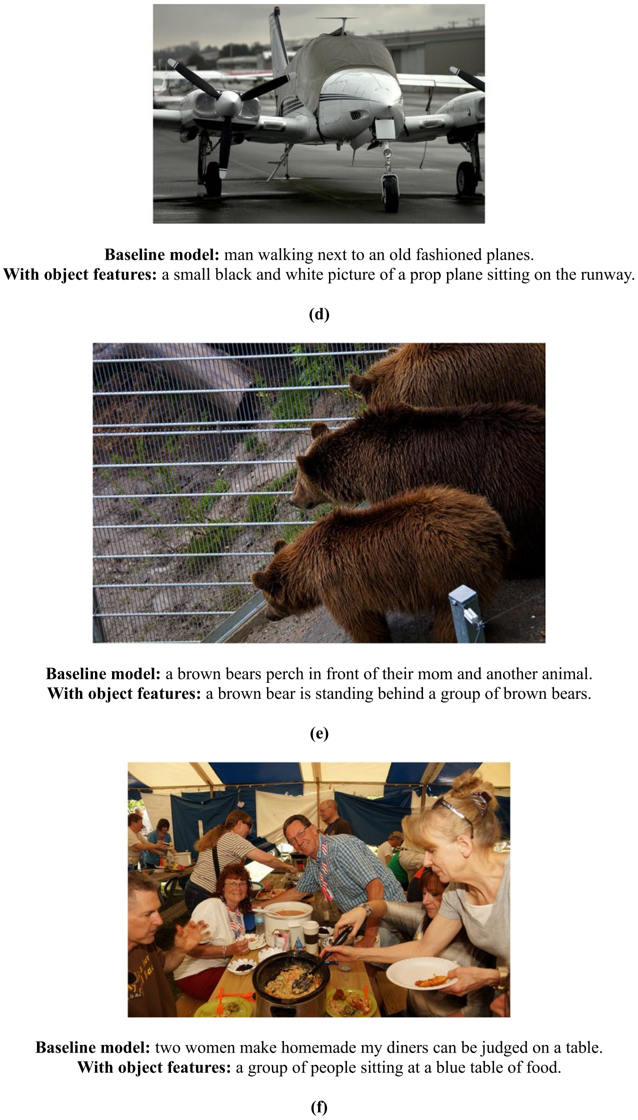
Baseline model: two people on skis sitting on a snowy surface.
With object features: a person standing next to snowboards attached.

(b)

Baseline model: a cow is standing in a open field as it grazes.
With object features: cows eat alone grazing on grasses in a hill.

(c)

Fig. 4 Qualitative examples from MS COCO comparing the generated captions before and after using our object features method, trained on MS COCO (with the importance factor). **a** Baseline model: this is up in snow pants jumping on a big snowy mountain at night. With object features: a skier performing a jump against some snow. **b** Baseline model: two people on skis sitting on a snowy surface. With object features: a person standing next to snowboards attached. **c** Baseline model: a cow is standing in a open field as it grazes. With object features: cows eat alone grazing on grasses in a hill. **d** Baseline model: man walking next to an old fashioned planes. With object features: a small black and white picture of a prop plane sitting on the runway. **e** Baseline model: a brown bears perch in front of their mom and another animal. With object features: a brown bear is standing behind a group of brown bears. **f** Baseline model: two women make homemade my diners can be judged on a table. With object features: a group of people sitting at a blue table of food



Baseline model: man walking next to an old fashioned planes.
With object features: a small black and white picture of a prop plane sitting on the runway.

(d)

Baseline model: a brown bears perch in front of their mom and another animal.
With object features: a brown bear is standing behind a group of brown bears.

(e)

Baseline model: two women make homemade my diners can be judged on a table.
With object features: a group of people sitting at a blue table of food.

(f)

Fig. 4 continued

with convolutional features. Future work can also benefit from rich object semantic information from caption texts instead of just object layouts, which can increase image captioning accuracy. Furthermore, more sophisticated methods can be used to encode object features before inputting them into the decoder, and more complex language models, such as Meshed-Memory Transformers [35] can be employed.

Abbreviations

CNN: Convolutional neural network; LSTM: Long short-term memory; GRU: Gated recurrent unit; VQA: Visual question answering; RPN: Region proposal network; IoU: Intersection-over-union; ACG: Anchor-centered graph; VSR: Verb-specific semantic roles; CIC: Controllable image captioning; GSRL: Grounded semantic role labeling; SSP: Semantic structure planner; LCS: Longest common subsequence.

Acknowledgements

Not applicable.

Authors' contributions

MAA-M: research, design, implementation and experiments, writing original version. AJ: supervision, validation, reviewing and editing. NG: supervision, validation, reviewing and editing. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The MS COCO dataset is available at: https://cocodataset.org/. The Flickr30k dataset can be requested here: https://shannon.cs.illinois.edu/DenotationGraph/.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

Not applicable.

Author details

[1]Present Address: Higher Institute for Applied Sciences and Technology, Syria, Damascus. [2]Arab International University, Syria, Daraa.

Received: 28 August 2021 Accepted: 30 January 2022
Published online: 14 February 2022

References

1. Farhadi A, Hejrati M, Sadeghi MA, Young P, Rashtchian C, Hockenmaier J, Forsyth D. Every picture tells a story: generating sentences from images. In: European conference on computer vision. Berlin: Springer; 2010. p. 15–29.
2. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.
3. 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. https://arxiv.org/abs/1406.1078. Accessed 3 Jun 2014.
4. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y. Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning. New York: PMLR; 2015. p. 2048–57.
5. Katiyar S, Borgohain SK. Image captioning using deep stacked LSTMs, contextual word embeddings and data augmentation. https://arxiv.org/abs/2102.11237. Accessed 22 Feb 2021.
6. Redmon J, Farhadi A. Yolov3: an incremental improvement. https://arxiv.org/abs/1804.02767. Accessed 8 Apr 2018.
7. Bochkovskiy A, Wang CY, Liao HY. Yolov4: optimal speed and accuracy of object detection. https://arxiv.org/abs/2004.10934. Accessed 23 Apr 2020.
8. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2017. p. 7263–71.
9. Yin X, Ordonez V. Obj2text: generating visually descriptive language from object layouts. https://arxiv.org/abs/1707.07102. Accessed 22 Jul 2017.
10. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556. Accessed 4 Sep 2014.
11. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2009. p. 248–55.
12. Vo-Ho VK, Luong QA, Nguyen DT, Tran MK, Tran MT. A smart system for text-lifelog generation from wearable cameras in smart environment using concept-augmented image captioning with modified beam search strategy. Appl Sci. 2019;9(9):1886.
13. 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;28:91–9.
14. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2016. p. 770–8.
15. Lanzendörfer L, Marcon S, der Maur LA, Pendulum T. YOLO-ing the visual question answering baseline. Austin: The University of Texas at Austin; 2018.
16. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2016. p. 2818–26.
17. Herdade S, Kappeler A, Boakye K, Soares J. Image captioning: transforming objects into words. https://arxiv.org/abs/1906.05963. Accessed 14 Jun 2019.
18. Wang J, Madhyastha P, Specia L. Object counts! bringing explicit detections back into image captioning. https://arxiv.org/abs/1805.00314. Accessed 23 Apr 2018.
19. Sharif N, Jalwana MA, Bennamoun M, Liu W, Shah SA. Leveraging Linguistically-aware object relations and NASNet for image captioning. In: 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). Piscataway: IEEE; 2020. p. 1–6.
20. Variš D, Sudoh K, Nakamura S. Image captioning with visual object representations grounded in the textual modality. https://arxiv.org/abs/2010.09413. Accessed 19 Oct 2020.

21. Alkalouti HN, Masre MA. Encoder-decoder model for automatic video captioning using yolo algorithm. In: 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS). Piscataway: IEEE; 2021. p. 1–4.
22. Ke A, Ellsworth W, Banerjee O, Ng AY, Rajpurkar P. CheXtransfer: performance and parameter efficiency of ImageNet models for chest X-Ray interpretation. In: Proceedings of the Conference on Health, Inference, and Learning. Harvard: CHIL; 2021. p. 116–24.
23. Xu G, Niu S, Tan M, Luo Y, Du Q, Wu Q. Towards accurate text-based image captioning with content diversity exploration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE; 2021. p. 12637–46.
24. Chen L, Jiang Z, Xiao J, Liu W. Human-like controllable image captioning with verb-specific semantic roles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE; 2021. p. 16846–56.
25. Cornia M, Baraldi L, Cucchiara R. Show, control and tell: a framework for generating controllable and grounded captions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE; 2019. p. 8307–16.
26. Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: a method for automatic evaluation of machine translation. In: Proceengs of the 40th annual meeting of the Association for Computational Linguistics. Philadelphia: ACL; 2002. p. 311–8.
27. Banerjee S, Lavie A. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. Philadelphia: ACL; 2005. p. 65–72.
28. Lin CY. Rouge: a package for automatic evaluation of summaries. In: Text summarization branches out. Barcelona: Association for Computational Linguistics; 2004. p. 74–81.
29. Vedantam R, Lawrence Zitnick C, Parikh D. Cider: consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2015. p. 4566–75.
30. Anderson P, Fernando B, Johnson M, Gould S. Spice: semantic propositional image caption evaluation. In: European conference on computer vision. Cham: Springer; 2016. p. 382–98.
31. Chollet F. Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2017. p. 1251–8.
32. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. https://arxiv.org/abs/1409.0473. Accessed 1 Sep 2014.
33. Karpathy A, Fei-Fei L. Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway: IEEE; 2015. p. 3128–37.
34. Plummer BA, et al. Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. Piscataway: IEEE; 2015.
35. Cornia M, Stefanini M, Baraldi L, Cucchiara R. Meshed-memory transformer for image captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE; 2020. p. 10578–87.

Publisher's Note

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