A NLP and YOLOv8-Integrated Approach for Enabling Visually Impaired Individuals to Interpret Their Environment

Roberta Avanzato[1], Lorenzo Mandelli[2,3] and Cristian Randieri[4]

[1]*Department of Electrical, Electronic and Computer Engineering University of Catania, Catania, Italy*
[2]*Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25, Roma, 00185, Italy*
[3]*ficonTEC Service GmbH, Im Finigen 3, 28832 Achim, Germany*
[4]*e-Campus University, Via Isimbardi, 10, 22060 Novedrate (CO), Italy*

Abstract

Image captioning represents a significant challenge within the field of Computer Vision. This task involves processing an image as input, identifying objects within it, comprehending the relationships between these objects, including their implicit characteristics, and generating a concise description as output. Given the vast number of potential interactions, acquiring sufficient training examples is a formidable task. Prior research has demonstrated that objects and predicates, when involved in less common relationships, occur more frequently when considered independently. Consequently, the proposed solution involves training two distinct visual models for objects and predicates, which are subsequently combined to capture as many relationships as possible. This study aims to leverage the information obtained regarding detected objects and their relationships to generate a comprehensive description of the image. By facilitating user interaction through Visual Question Answering, a task that bridges Computer Vision and Natural Language Processing, we can create an interactive approach to image captioning. Given a question and an image, the system is designed to reason based on both the image content and general knowledge, and generate an accurate answer. We believe that such a system provides an effective method for visually impaired individuals to understand the content of an image.

Keywords

NLP, YOLOv8, Support Systems, Automatic Classification, Convolutional Neural Networks

1. Introduction

Assistive technologies strive to enhance the quality of life for individuals with visual impairments by facilitating access to various forms of content. This paper concentrates on the development of a system capable of interpreting photographs. Determining the most effective method for describing an image is a complex task, even when performed by humans. There are no universal guidelines for deciding which critical information should be emphasized, particularly when the objective is to render an image accessible to a visually impaired person. Often, in an attempt to provide a more detailed understanding, one might resort to drawing parallels closely related to personal experiences or preferences. This approach not only results in a highly subjective description but also risks creating divergent perceptions between the describer and the listener, effectively transforming a single image into two distinct interpretations. The implementation of a system to achieve this objective would undoubtedly necessitate the provision of a more objective description. The only experiential basis we can rely on is the frequency with which our network identifies a pair of subject and object linked by a specific predicate. Machine learning has revolutionized the field of scientific research in recent years. It has enabled researchers to process and analyze large datasets, leading to new discoveries and insights. Machine learning algorithms can identify patterns and correlations in data that may not be immediately apparent [1]. This has been particularly impactful in fields such as neuroscience, climate science [2], and human machine interaction [3], where the volume of data can be overwhelming [4, 5], as well to support people with impairments, as well asl physical or mental ilnesses or difficulties of various kind [6, 7, 8, 9]. Additionally, machine learning is being used to predict outcomes and trends, aiding in hypothesis generation and testing. However, the use of machine learning in scientific research also presents challenges, particularly in terms of ensuring the transparency and reproducibility of results. Upon receiving an image as input, we propose a visual modeling approach aimed at discerning the visual relationships between depicted objects. These relationships are elucidated through a scene graph (depicted in Figure 1), a graph-based data structure wherein nodes represent objects and edges denote predicates. This structured representation of the scene is subsequently stored in a Knowledge Base, formatted as $\langle subject - predicate - object\rangle$. Subsequently, user interaction is facilitated, offering brief descriptions of the depicted image and enabling further inquiry through Visual Question Answering tasks. The initial phase involves developing a model capable of discerning object relationships within an image. This entails separate training for both object and predicate recognition, followed by their integration to infer relationships [10]. Prior studies have demonstrated the efficacy of learning these components individually due to their frequent occurrence in isolation. Consequently, the model can infer specific relationships not explicitly encountered by leveraging the semantics of more commonly observed relationships. In our approach, we employ YOLO (You Only Look Once) [11] for object recognition, while constructing a dedicated training model solely for relationship predictio YOLO represents a cutting-edge real-time object detection system acclaimed for its exceptional speed and accuracy. However, it occasionally presents a discrepancy in the number and labels of detected objects. This challenge is addressed by aligning YOLO classes with the most analogous ones anticipated by our network. Our experimentation began with a baseline model trained solely on visual and spatial features, which was subsequently augmented by integrating a language model [12] leveraging word embeddings. The subsequent phase involves training a model capable of responding to questions based on the visual content of an image. However, it is conceivable that the sought-after answer may be independent of the image

SYSYEM 2023: 9th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Rome, December 3-6, 2023

roberta.avanzato@unict.it (R. Avanzato); mandelli@diag.uniromai.it (L. Mandelli); cristian.randieri@uniecampus.it (C. Randieri)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

and necessitate information not present within it. Consequently, it is imperative to construct a Knowledge Base. This serves the dual purpose of preserving extracted information in an accessible format and integrating diverse sources of knowledge. As previously noted, Visual Question Answering (VQA) has garnered attention from both the Computer Vision and Natural Language Processing domains. Tasked with providing correct answers given an image and a question, the VQA module's experimentation commenced with a baseline model trained solely on image feature extraction and question encoding. Subsequently, we incorporated a parallel co-attention model [13] to identify image regions pertinent to answer prediction. In addressing questions, we adhere to established protocols for text data handling, including contraction resolution, tokenization, padding for uniform sentence lengths, and word embedding for semantic representation, whereby words closer in the vector space denote similar meanings. Upon answer generation, typically in written format for this task, the final step entails integrating a text-to-speech module to facilitate auditory communication of the obtained information to the user. The system exhibits favorable performance when presented with images of moderate complexity.

2. Related Work

Utilizing neural networks for generating image descriptions represents an intriguing intersection of image understanding methodologies and natural language processing techniques. A seminal study by Lu et al. [10] presents a model that trains separate visual models for objects and predicates. Specifically, a Convolutional Neural Network (CNN), such as VGG net, is employed to classify objects, while a second VGG net is utilized to identify predicates based on the union of bounding boxes of the interacting objects. These models are then integrated to predict a diverse range of visual relationships within each image. The rationale behind this approach is rooted in the semantic relatedness of relationships: by integrating a language module, the model is primed to select the appropriate relationship more accurately. As relationships that occur frequently in the training data are more likely to recur, they are easier to infer. However, even if a relationship has not been previously encountered, if it shares semantic similarities with known relationships, the model should still be able to comprehend it. To facilitate this, the two objects involved in a relationship are projected into a word embedding space using a pre-trained word vector (e.g., word2vec) and concatenated to form the relationship vector space. The projection function is then optimized to ensure that similar relationships are situated close together; the distance between relationships is proportional to the word2vec distance between their constituent objects and predicate. Furthermore, Lu et al. introduced a Visual Relationship Detection (VRD) dataset, comprising 5000 images featuring 100 object categories and 70 predicates categorized into four types: verbs, prepositions, spatial relations, and comparatives. This dataset serves as the foundation for our experimental endeavors. Jung et al. [12] further refined the previously described model by training the language module using softmax loss instead of the Kullback-Leibler (K, L) loss. They observed that the Kullback-Leibler loss function encourages similar visual relationships to converge while pushing dissimilar relationships apart. However, even in the absence of this loss function, word vectors naturally tend to cluster similar visual relationships together. Additionally, the L loss function assigns higher likelihood to high-frequency data and lower likelihood to low-frequency data, leading to a more natural distribution of likelihoods for predicates based on their frequency. Furthermore, the integration of the language and visual modules was modified to include a point-wise multiplication between them. Subsequently, the image, presumed to contain all necessary information, along with a textual question, serves as the starting point for Visual Question Answering (VQA) [14]. Typically, the sought-after answer in VQA tasks is a brief phrase or a few words. In some approaches, Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cells are employed. RNNs are adept at processing both questions and answers of variable lengths. The question and image features, the latter derived from a pre-trained Convolutional Neural Network (CNN) for object recognition, are jointly input to an encoder LSTM. The resulting fixed-size feature vector is then passed to a decoder LSTM for answer generation. This process can be framed as a sequence generation task or treated as a classification problem in some instances. Alternatively, bidirectional LSTMs are utilized to capture relationships between distant words in the question more effectively. Alternatively, CNNs are employed to process questions, with features from both the image and text embedded in a shared space. In contrast to approaches that simply concatenate image and question vectors, Lu et al. [13] proposed an Attentional Network aimed at generating a more informative query vector. This network comprises three key steps: employing a VGG net to extract image representations, utilizing LSTMs and CNNs to capture the semantic nuances of textual input, and incorporating a parallel co-attention layer for answer prediction. Specifically, in the final step, image features are combined with a three-level question representation to compute the attention distribution over different regions of the image. The dataset employed in their study is COCO-QA, which comprises 123,287 images, along with 78,736 training questions and 38,948 test questions categorized into four types: object, number, color, and location. Answers to these questions are all single-word responses, consistent with the format used in our experiments. However, the structure of the questions varies, and extracting ⟨subject - predicate - object⟩ triplets from a sentence is typically achieved [15] using a parser that constructs a tree rooted in the subject (S) and with three children: a noun phrase (NP), a verbal phrase (VP), and punctuation. The subject of the sentence is typically found within the NP subtree, while the predicate is identified as the deepest verb descendant within the VP subtree. Objects are typically located within prepositional phrase (PP), noun phrase (NP), or adjective phrase (ADJP) subtrees, specifically as the first noun or adjective encountered. Leveraging this parsing mechanism facilitates the establishment of a connection between the VQA model and the knowledge base.

3. The proposed method

In Figure 2, we present the architectural overview of our interactive image captioning system. Specifically, the relationship detection module is responsible for extracting comprehensive information from the image, while the knowledge base, serves as a repository for storing this information. Subsequently, users engage with the system through an in-

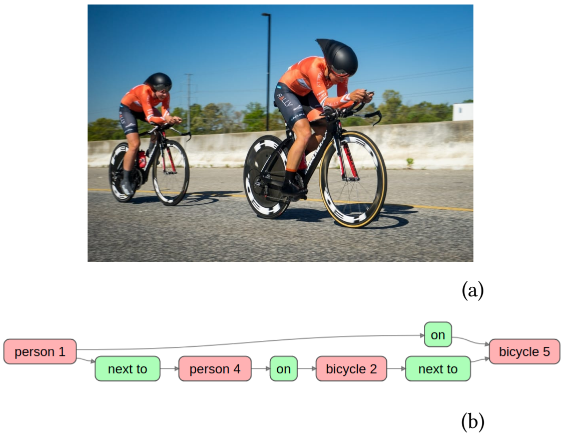


Figure 1: An example of scene graph extraction using our Relationship Detection Model: (a) the input image; (b) the scene graph of the image

teractive module, allowing them to receive explanations or pose questions. Responses to these inquiries are provided either by querying the knowledge base or through the Visual Question Answering module. Subsequent sections offer a detailed exposition of each of these modules.

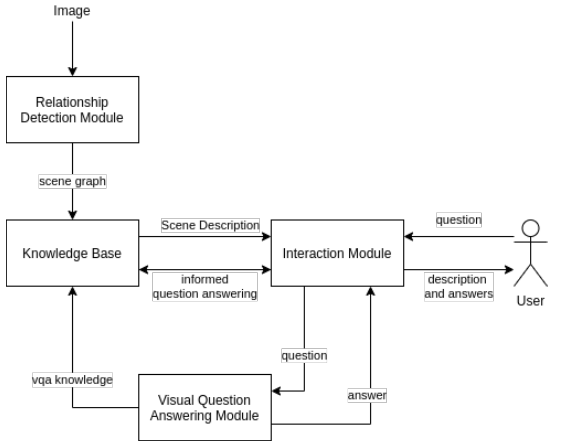


Figure 2: Our system structure

As previously mentioned, the relationship detection module encompasses the fusion of two distinct systems: one dedicated to object detection and the other to relationship identification. YOLO (You Only Look Once) [11] represents a neural network architecture designed to predict bounding boxes and class probabilities across entire images, implicitly incorporating contextual information. Renowned for its exceptional speed, YOLO surpasses other real-time systems by achieving more than twice the mean average precision. Additionally, YOLO demonstrates a capacity to learn generalizable object representations, enhancing its resilience when applied to novel domains. In the context of our discussion, bounding boxes denote regions within images encompassing potential objects, specified by coordinates in the format: $[x_{min}, y_{min}, width, height]$. We leverage YOLOv3 to extract objects from images, subsequently utilizing its output to establish relationships between them, as elaborated upon in the subsequent module description. Initially, employing YOLO for object detection inevitably introduces a discrepancy in class labels between YOLO's 80 classes and the 100 classes expected by our network. Consequently, it becomes imperative to establish associations between the most similar labels to mitigate this discrepancy. Subsequently, we turn our attention to the computation of spatial features within an image. Instead of utilizing a spatial vector solely encoding the normalized location and size of each object's bounding box within an image, we adopt a methodology proposed by Jung et al. [12]. This method incorporates the Intersection Over Union (IOU) and normalized relative location (x, y) based on the subject box center, as well as normalized subject and object sizes. Additionally, it includes a containment flag for both the subject and object, where the subject flag equals 1 if the subject box contains the object box, and 0 otherwise. Here, bounding boxes are represented as $[y_{min}, y_{max}, x_{min}, x_{max}]$. Computing the IOU involves dividing the area of overlap between bounding boxes by the area of their union. In the baseline configuration, visual features from the sub-image are extracted using a VGG16 model pre-trained on ImageNet and concatenated with spatial features. The sub-image is crafted to encompass both the subject and object. Original images are rescaled to dimensions of 256×256 pixels, followed by extraction of a $512 \times 7 \times 7$ feature map using the VGG net. This feature map is then flattened into a one-dimensional vector and transformed into a 1×1024 vector through a series of fully connected layers. Subsequently, this feature vector is combined with the spatial features and utilized as input for a classifier to generate the final prediction. Upon integration of a language module, relationships are projected into an embedding space to facilitate recognition of similar relationships. Utilizing the simplified FastText open-source library, word embedding and text classification are achieved. Specifically, only the 100 objects present in the dataset are considered, as the entire FastText vocabulary is excessively large for memory constraints. Subsequently, another VGG16 model is employed to extract a $512 \times 7 \times 7$ feature map, which is then resized into a 1×2048 vector through fully connected layers. These features are concatenated with spatial features and transformed into a 1×1024 vector. Following this, word embedding is extracted and element-wise multiplication is performed between concatenated word features and spatial/image features. The resultant features are utilized for relation prediction through a linear classifier. The comprehensive model architecture is depicted in Figure 3. To prevent duplicate relationships involving the same pair of objects, the scene graph is post-processed. Specifically, for each pair of relationships, if the subject of one pair matches the object of the next pair and vice versa, only the more relevant relationship is retained. Relevance is determined by computing the probability of encountering the subject-object pair in the training set. The relationship with the higher occurrence probability is selected, ensuring inclusion of the relationship on which the model has more training data and thus greater confidence. For example, if the relationships $\langle person\_0 - on - horse\_1 \rangle$ and $\langle horse\_1 - next\_to - person\_0 \rangle$ are encountered, the probabilities of the subject-object pairs "person-horse" and "horse-person" are computed based on their occurrences in the training set. The more relevant relationship, such as "person-horse" in this scenario, is retained, resulting in the inclusion of only the $\langle person\_0, on, horse\_1 \rangle$ relation in the scene graph. This approach ensures that relationships with greater training data support are prioritized, enhancing the model's confidence in its predictions.

Another problem faced with the scene graph, is the choice

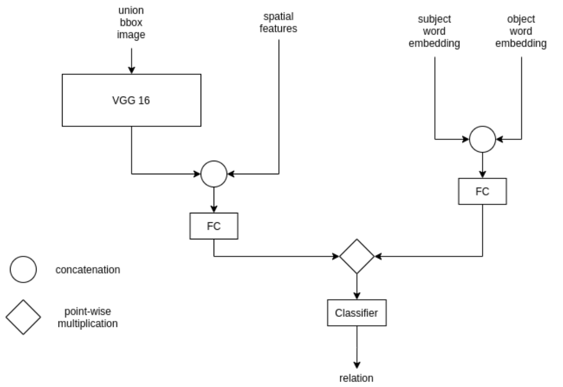


Figure 3: Our Relationship Detection Module

of which objects couple in the image is good to be taken in consideration to compute the relation. If all the objects couple are taken, the number of generated relationship is exponential in terms of elements in the scene. For this reason, it is useful to filter out some couple. We approach this issue by computing the normalized (with respect to the image size) distance between the centers of the bounding boxes for each pair of them. If the normalized distance is lower than a threshold (that we choose to be 0.12, with empirical tests), then the couple is considered to be valid, otherwise it is discarded. In this way, only near enough objects are considered to be in relation, and we avoid the explosion of relations with complex images. The Visual Relationship Detection (VRD) dataset consists of 5000 images featuring 100 distinct object categories and 70 predicates. Specifically, the dataset comprises 37,993 relationships spanning 6,672 unique relationship types, with an average of 24.25 predicates associated with each object category. The training set comprises 4000 images, while the remaining 1000 images constitute the test set. Notably, 1,877 relationships are unique to the test set and are absent from the training set. To optimize training efficiency, we adopt a selective approach in predicate selection. Among the 70 predicates provided by the dataset, we prioritize those exhibiting a higher frequency of occurrence, specifically selecting predicates that appear more than 75 times in the training set. This strategy aims to foster focused learning within the network, enhancing its capacity to discern and generalize patterns from the training data. In the Table 1, the improvement obtained with the addition of the language is highlighted with respect to the baseline, by the clear increase in accuracy value achieved. As observed, the accuracy metric, while not egregiously low, does not elicit enthusiasm. This observation aligns with findings in the literature [10], which underscore the challenging nature of Visual Relationship Detection, despite its relative simplicity for humans. Nonetheless, the modest accuracy warrants a comprehensive investigation. Upon scrutinizing the dataset, it becomes apparent that certain classes are underrepresented, with limited occurrences of some relationships. Consequently, the overall accuracy suffers, reflecting the challenges posed by data sparsity. However, an examination of the macro-precision score (refer to Table 2) reveals promising results for larger classes, indicating satisfactory performance in these instances. One potential approach to address the observed low performance is the adoption of more complex network configurations, such as Graph Neural Networks, along with leveraging larger datasets. However, due to constraints imposed by limited computational resources, these approaches are not feasible within the scope of this study.

	baseline	with language
VRD	33.38%	**63.01**%

Table 1
Relationship Detection accuracy results

	baseline	with language
VRD	65.30%	**81.90**%

Table 2
Relationship Detection precision results

Subsequently, the performance of the other component of the system, YOLO-v8, pre-trained on the COCO dataset, is assessed using the same dataset employed for the Relationship Detection Model. Evaluation of the YOLO model for object detection is conducted using the Intersection over Union (IoU) metric. IoU quantifies the degree of overlap between two bounding boxes: one representing the ground truth and the other corresponding to the predicted bounding box. A IoU value of 1 indicates perfect overlap between the two bounding boxes.

To determine the validity of object detection, a threshold of 0.5 is set for IoU. If the IoU value is greater than or equal to the threshold, the object detection is classified as True Positive (TP); otherwise, it is classified as False Positive (FP) indicating an erroneous detection. Moreover, instances where a ground truth object is present in the image but remains undetected by the model are classified as False Negative (FN).

Subsequently, the precision metric is computed using the conventional formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Table 3 presents the precision results for both the VRD model and the YOLO model, juxtaposing their performances to underscore the influence of each component on the overall task performance.

Notably, the evaluation reveals that YOLO, pre-trained on the COCO dataset, does not yield satisfactory performance for this task when applied to a distinct dataset such as the one used for Visual Relationship Detection.

	precision
VRD with language	81.90%
YOLO-v8	77.31%

Table 3
Precision results for our best Relationship Detection model (VRD) and for YOLO model used for object detection

The foundation of our system lies in a Knowledge Base (KB), designed to fulfill several key functions: (1) recording information provided by the Relationship Detection module; (2) capturing insights gained from user interaction during the execution of the Visual Question Answering (VQA) module; and (3) facilitating the VQA Module in generating responses. The Resource Description Framework

(RDF), proposed by the World Wide Web Consortium (W3C), serves as the underlying framework for storing and sharing knowledge across the web. We leverage this framework to construct our knowledge base. It's important to note that we refrain from establishing connections with web URIs of RDF concepts, opting instead for a fully localized knowledge base. In this subsection, we elaborate on how RDF triplets are utilized to fulfill the aforementioned tasks. Upon extraction of the scene graph from the image by the Relationship Detection module, all triplets are stored in the knowledge base. Additionally, the knowledge base is initialized with a rudimentary, manually crafted prior knowledge structure, encompassing all YOLO classes. The structure of this prior knowledge is illustrated in Figure 4, serving as a demonstration of how a more intricate graph can be constructed to enhance the system's performance in specific scenarios.

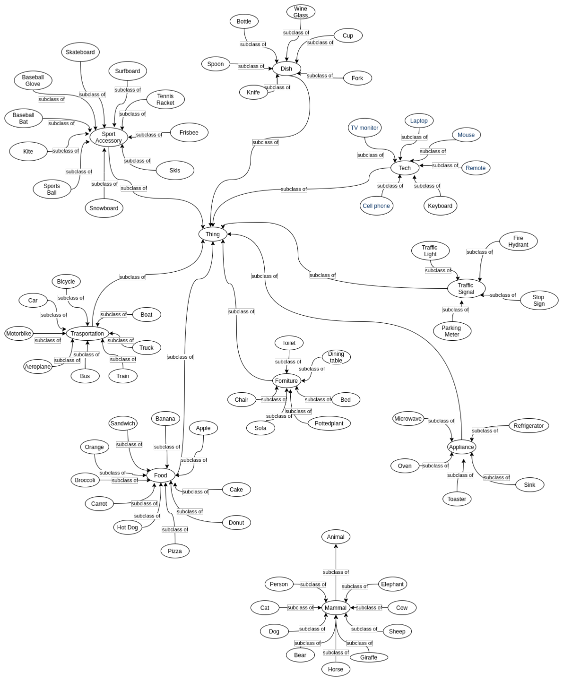
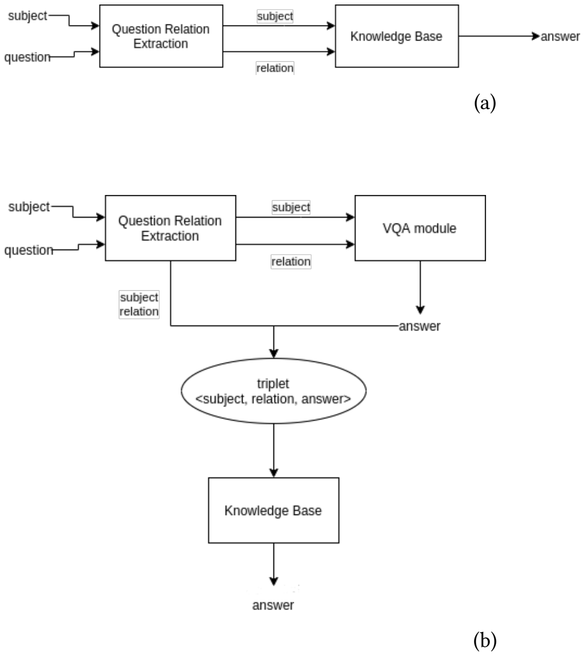


Figure 4: The Structure of our Prior Knowledge Base

Once all objects within the scene are linked to the prior knowledge graph via an "is_a" relation (for instance, if a dog is present in the image with an identifier of 1, the triplet $\langle dog\_1 - is\_a - Dog\rangle$ is appended), all object-to-object relationships inferred by the preceding module are incorporated into the knowledge base (KB). The primary role of the RDF triplets is to aid the VQA model. The pipeline utilized to support the VQA model with the knowledge base is depicted in Figures 5a and 5b. These figures outline the algorithmic process employed to leverage the information stored within the knowledge base for VQA tasks.



(a)

(b)

Figure 5: Answer Support Pipeline when: (a) the KB knows the answer; (b) the KB does not know the answer

For each question in which the subject is implicit, facilitated by the "focus mechanism" (refer to section **??**), the predicate is extracted to construct the relationship (for details on predicate extraction from questions and relationship construction, see the subsequent subsection). Subsequently, if a triplet in the form of $\langle subject - relation - object\rangle$ already exists in the knowledge base (KB), the object string is retrieved as the answer. This process enables the system to respond to questions for which relevant knowledge is already available in the KB, whether the answer was previously provided by the VQA module or included in the prior knowledge. In the absence of such a triplet, the VQA module is invoked, and the resultant answer is then stored in the knowledge base before being presented to the user. Adding the answer as new information involves extracting the relationship from the question and utilizing the answer as the object of the triplet. In [15], an algorithm for extracting triplets from sentences is introduced (refer to Figure **??** and **??**). We leverage this concept to devise an algorithm for extracting relationships from questions. Initially, a parse tree of the question is generated using the BLLIP parser [16]. Subsequently, we employ the "EXTRACT-PREDICATE" function to obtain the predicate, with slight modifications: we execute the function on the entire tree rather than solely the VP_subtree, given that questions may not consistently contain VP/NP subtrees. Additionally, we consider the PRT tag, not just the ADVP tag, to capture the attribute of the verb. Because our goal is to extract a relation, we also need a support for the predicate, in order to do not obtain the same relation string for questions with the same predicate. For example, if we consider the two following questions: "What is the color of this object?" and "What is the genre of this book?", we will obtain the same predicate and so the same relation. This can be a problem, because the two questions will be considered the same when we try to answer to the question with the KB module. For this reason, the actual relation is built with the predicate on which we append the "reference noun" of the sentence. This "reference noun" is chosen taking all the NN leafs of the parse tree. In the figure **??**, an example of relation extraction is illustrated. Given our objective of extracting relationships, it is essential to consider support for the predicate to avoid generating identical relation strings for questions sharing the same predicate. For instance, if we analyze the following questions: "What is the color of this object?" and "What is the genre of this book?", both questions share the predicate "is," potentially leading to the generation of identical relations. This situation poses a challenge, as the system may treat these questions as equivalent when attempting to provide answers using the KB module. To address this issue, the constructed relation incorporates the predicate augmented by the "reference noun" of the sentence. This "reference noun" is determined by considering all the NN leaf nodes of the parse tree. For the Visual Question Answering (VQA) model, we develop both a simple baseline and a model incorporating co-attention mechanism. This section provides a detailed explanation of both models, followed by a presentation of other attempted approaches. Additionally, we demonstrate the superior performance of the co-attention model over alternative methods. In the baseline approach, we employ the most straightforward strategy for VQA, as depicted in Figure 6. Initially, the image features are extracted using a pre-trained VGG16 model trained on the ImageNet dataset to capture the visual semantics of the image. Subsequently, the question is encoded using a one-hot encoding scheme and processed through an LSTM network to obtain the features of the question's words within the same semantic space as the image. Next, utilizing the vector representations of the image and the question, a point-wise multiplication is conducted to derive a feature vector that encapsulates the combination of both inputs. Finally, a prediction is made using a series of fully connected layers to determine the answer to the question posed. In this approach, while the network is adept at discerning relevant words for the task based on the image feature vector, the results are less than satisfactory. This is primarily due to the network's incapacity to pinpoint specific image regions crucial for accurate question answering. Notably, the model's performance falters when tasked with providing precise responses, particularly when answers depend on nuanced image regions. Consequently, integrating an attention mechanism becomes imperative. As previously discussed, effectively answering questions about an image requires understanding the relevant image regions containing pertinent information. To tackle this challenge, we propose employing a co-attention mechanism between the image and the question. The overarching architecture of our model, , draws significant inspiration from [13]. A crucial aspect of this methodology is considering not only "where to look" within the image to address the question but also "which words to attend to" from the question itself. By incorporating both aspects, our model aims to achieve a more comprehensive understanding of the question-image relationship, thereby enhancing its ability to generate accurate responses. Following the methodology outlined in [13], our system comprises three primary components: (1) the image model, which leverages a pre-trained VGG-16 network to extract high-level image representations, yielding one vector for each region; (2) the question model, employing a Long Short-Term Memory (LSTM) network to derive a semantic vector from the question, alongside a Convolutional Neural Network (CNN) to capture information across various levels; and (3) the co-attention model, responsible for identifying image regions and words pertinent to the question for subsequent answer prediction. The image features are extracted using a VGG-16 network pre-trained on the ImageNet dataset, subsequently fine-tuned for several epochs. Specifically, high-level features are extracted to generate a representation for each region of the image (refer to Figure 7). The images undergo resizing to dimensions of 448 × 448 pixels. Subsequently, the output of the final pooling layer of the VGG-16 network is obtained, resulting in a structured output of dimensions 512 × 14 × 14. This process yields 14 × 14 feature vectors, each with a dimensionality of 512, corresponding to the 196 regions, each spanning 32 × 32 pixels. With this representation, we can effectively compute attention on regions of interest by combining it with the question embedding. We construct a question representation at three levels: (a) word level, (b) phrase level, and (c) question level. At the word level, mirroring the baseline approach, the question undergoes one-hot encoding with a pre-trained FastText embedding layer, yielding a word-embedded representation of the question:

$$\mathbf{Q} = \{\mathbf{q}_1^w, ..., \mathbf{q}_T^w\} \tag{1}$$

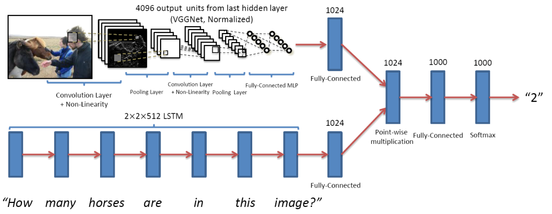


Figure 6: Baseline model

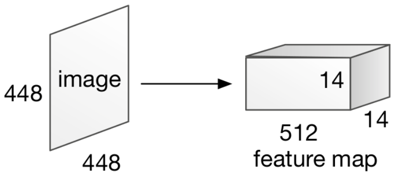


Figure 7: VGG-16 feature extraction [13]

At the phrase level, we apply a one-dimensional convolution on the word embedding vectors. Specifically, at each word location, we compute the inner product of the word vectors with filters of three window sizes: unigram, bigram, and trigram. For the t-th word, the convolution output with window size s is computed as:

$$\hat{\mathbf{q}}^p{}_{s,t} = tanh(\mathbf{W}_c^s \mathbf{q}_{t:t+s-1}^w), \quad s \in \{1, 2, 3\} \tag{2}$$

where $\mathbf{W}_c^s$ represents the weight parameters. Subsequently, we apply max-pooling across different n-grams at each word location to obtain phrase-level features:

$$\hat{\mathbf{q}}^p{}_t = max(\hat{\mathbf{q}}^p{}_{1,t}, \hat{\mathbf{q}}^p{}_{2,t}, \hat{\mathbf{q}}^p{}_{3,t}), \quad t \in \{1, 2, ..., T\} \tag{3}$$

At the question level, we pass $\hat{\mathbf{q}}^p{}_t$ through a Long Short-Term Memory (LSTM) unit and extract the hidden vector at time t. This process enables us to capture the information encompassed in the entire question. The co-attention model, inspired by the Parallel Co-Attention mechanism in [13], is pivotal in our system. It involves computing the similarity between image and question features across all pairs of image-locations and question-locations. Beginning with an image feature map $\mathbf{V} \in \mathbb{R}^{d \times N}$ and the question representation $\mathbf{Q} \in \mathbb{R}^{d \times T}$, we calculate the affinity matrix $\mathbf{C} \in \mathbb{R}^{T \times N}$ as follows:

$$\mathbf{C} = \tanh(\mathbf{Q}^T \mathbf{W}_b \mathbf{V}) \tag{4}$$

Here, $\mathbf{W}_b \in \mathbb{R}^{d\times d}$ represents the weight parameters. Subsequently, we use this affinity matrix to predict image and question attention maps:

$$\mathbf{H}^v = \tanh(\mathbf{W}_v\mathbf{V} + (\mathbf{W}_q\mathbf{Q})\mathbf{C}) \quad (5)$$

$$\mathbf{H}^q = \tanh(\mathbf{W}_q\mathbf{Q} + (\mathbf{W}_v\mathbf{V})\mathbf{C}^T) \quad (6)$$

$$\mathbf{a}^v = \text{softmax}(\mathbf{w}_{hv}^T\mathbf{H}^v) \quad (7)$$

$$\mathbf{a}^q = \text{softmax}(\mathbf{w}_{hq}^T\mathbf{H}^q) \quad (8)$$

Here, $\mathbf{W}_v, \mathbf{W}_q \in \mathbb{R}^{k\times d}$, $\mathbf{w}_{hv}, \mathbf{w}_{hq} \in \mathbb{R}^k$ are the weight parameters, and $\mathbf{a}^v \in \mathbb{R}^N$ and $\mathbf{a}^q \in \mathbb{R}^T$ represent the attention probabilities for each image region $\mathbf{v}_n$ and word $\mathbf{q}_t$, respectively. The attention-weighted image and question features are then computed as weighted sums:

$$\hat{\mathbf{v}} = \sum_{n=1}^{N} a_n^v \mathbf{v}_n, \quad \hat{\mathbf{q}} = \sum_{t=1}^{T} a_t^q \mathbf{q}_t \quad (9)$$

This parallel co-attention process is conducted at each level of the hierarchy, resulting in $\hat{\mathbf{v}}^r$ and $\hat{\mathbf{q}}^r$, where $r \in \{w, p, s\}$. Finally, the answer is predicted based on the co-attended image and question features from all three levels, employing a multi-layer perceptron to recursively encode the attention features:

$$\mathbf{h}^w = \tanh(\mathbf{W}_w(\hat{\mathbf{q}}^w + \hat{\mathbf{v}}^w)) \quad (10)$$

$$\mathbf{h}^p = \tanh(\mathbf{W}_p[(\hat{\mathbf{q}}^p + \hat{\mathbf{v}}^p), \mathbf{h}^w]) \quad (11)$$

$$\mathbf{h}^s = \tanh(\mathbf{W}_s[(\hat{\mathbf{q}}^s + \hat{\mathbf{v}}^s), \mathbf{h}^p]) \quad (12)$$

$$\mathbf{p} = \text{softmax}(\mathbf{W}_h\mathbf{h}^s) \quad (13)$$

Here, $\mathbf{W}_w, \mathbf{W}_p, \mathbf{W}_s$ and $\mathbf{W}_h$ are the weight parameters, $[\cdot]$ denotes concatenation, and $\mathbf{p}$ represents the probability distribution over all possible answers. Drawing inspiration from [17], our first approach involves augmenting the word embedding vector with two additional embeddings: lemma embedding and POS embedding. The lemma embedding is generated using a simple embedding layer for the lemmas of the words in the question, while the POS embedding is obtained by POS-tagging the words and utilizing the resulting tags to produce the embedding vector. This integration allows for the inclusion of both the POS-tag information and lemma information within the question embedding. In contrast, the second approach explores the concept of stacked attention layers [18]. Here, we utilize the attention map $\mathbf{H}^q$ to generate an attention-weighted question through pointwise multiplication:

$$\hat{\mathbf{q}}_t^p = \mathbf{H}^q * \hat{\mathbf{q}}_t^p \quad (14)$$

Subsequently, we leverage the resulting attention-weighted question for a second co-attention layer. Despite these attempts, neither of these methods yield improvements to the model, as demonstrated in Table **??**. Further details are omitted due to their lack of efficacy.

4. Results

We assess the proposed models using two datasets: the VizWiz dataset [19] and the COCO-QA dataset [20]. **VizWiz** comprises both single-word and multi-word answers. It consists of 20,523 training image/question pairs, 205,230 training answer/answer confidence pairs, 4,319 validation image/question pairs, 43,190 validation answer/answer confidence pairs, and 8,000 test image/question pairs. Given our requirement for single-word answers and the ability to use the accuracy metric to gauge performance, we filter out all multi-word answers from the dataset. Additionally, since each question/answer pair includes ten proposed answers, we select the first word that does not have an "unanswerable" label with a "yes" or "maybe" confidence score. We designate "unanswerable" only if all ten proposals carry this label, aiming to minimize noise in the answers. Despite these measures, the dataset's performance remains subpar due to the presence of numerous "noisy questions" incompatible with our project's scope. Consequently, we opt to utilize weights trained on the COCO-QA dataset for the VQA model integrated into the final system. COCO-QA is a dataset exclusively comprising single-word answers, encompassing 78,736 training samples and 38,948 test samples.

5. Case study

Before the interaction commences, the system initially presents the user with a list of all objects in the scene, followed by an explanation of the relationships between them (comprising all the triplets extracted from the Relationship Extraction module). Subsequently, the interaction begins, and the system enters state 1. Here, the user has the option to request a repeat of the object list or the interaction details between them. Additionally, the user can utilize the "focus mechanism" to choose an object of interest within the scene. The "focus mechanism" draws inspiration from the natural exploratory behavior of the eyes when examining an image. Initially, the eyes survey the entire picture to gain an overview before focusing on specific objects to scrutinize details of interest. In this interactive system, users can autonomously select the object they wish to focus on and explore specific details through questions. Essentially, the system awaits a request to focus on a particular object within the scene. Each object is assigned a unique identifier. If the chosen object is the only one of its category in the image, the user can simply state the object's label. Otherwise, the user must first specify the label, after which the system assists in disambiguating the object of interest. For instance, the user may initially state: "I want to focus on a person," prompting the system to inquire about the id of the desired person, listing all available options for selection. Once the focus is established, the system transitions to state 2, enabling the user to pose questions regarding the subject of interest (explicitly identified in the creation of triplets inserted into the Knowledge Base. It is crucial to note that the sub-image contained within the bounding box of the subject of interest is utilized as input for the VQA model, rather than the entire image. Upon request, the focus can be altered, reverting the system back to state 1. Initially, the system prompts the user to wait while preliminary image extractions occur. Subsequently, the system elucidates the relationships between objects in the image and awaits user commands. The user selects a focus, transitioning to state 2. Modularity is an important feature of our system and in this section is presented the way in which the system can be expanded to extract more information from the image in order to improve the user experience. In particular, there are two approaches for modularity: (1) new features can be extracted from the whole image, in order to better give a general overview or (2) new modules can be added to support the VQA model to accommodate the use of the system in specialized environments. To show how this mechanism works we have developed two new neural models, one for each kind of modularity. Below the two networks are presented, then the way in which the modularity is integrated in the implementation is illustrated. With this module, our aim is to determine in which place the image is set, in order to give a very useful information about the environment represented in the image to the user. We believe that this improves considerably the understanding of the image. For this scope, that is to show how the system can be improved adding modules, we use a simple and light dataset: 15-Scene Dataset [21]. It contains 4,485 gray-scale images of 15 scene categories (see Table 4): 5 indoor and 10 outdoor. Each class contains from 210 to 410 scene images, and the image size is about 300 × 250. We use 80% of the dataset for train and 20% for validation (the number of images, per class, is showed in Table 4).

	Scene Class	# train samples	# test samples
0	bedroom	170	46
1	city	254	54
2	coast	294	66
3	forest	268	60
4	highway	213	47
5	industrial	244	67
6	kitchen	163	47
7	living room	232	57
8	mountain	302	72
9	office	175	40
10	open country	324	86
11	store	253	62
12	street	242	50
13	suburban	189	52
14	tall building	265	91
	TOTAL	3588	897

Table 4
15-Scenes Dataset Statistics

We can affirm that with this simple model, easily integrated in the modular mechanism adopted by our system, the user experience can be considered improved and can encourage to add other module to extract image features useful for understanding. Modularity serves as a cornerstone of our system's design, facilitating its extensibility to extract additional information from images, thereby enhancing the user experience. In this section, we explore two avenues for modularity: (1) the extraction of new features from the entire image to provide a comprehensive overview, and (2) the addition of new modules to bolster the VQA model's capabilities, catering to specialized environments. To demonstrate this mechanism, we have developed two new neural models, each catering to one aspect of modularity. The subsequent sections detail these networks, followed by an illustration of how modularity is seamlessly integrated into the implementation. The first module aims to determine the location depicted in the image, thereby providing valuable contextual information about the depicted environment to the user. We anticipate that this addition significantly enhances image comprehension. For demonstration purposes, showcasing the system's capacity for enhancement through modular additions, we utilize a lightweight dataset: the 15-Scene Dataset [21]. Comprising 4,485 grayscale images spanning 15 scene categories (as delineated in Table 4), the dataset encompasses a diverse array of indoor and outdoor scenes. Each class contains between 210 to 410 scene images, with an average image size of approximately 300 × 250 pixels. We partition 80% of the dataset for training and reserve 20% for validation, as outlined in Table 4. To facilitate the seamless integration of new modules into the system, we have devised an extension module. As previously elucidated, there are two conceivable approaches to extension. The first involves implementing a module capable of extracting information from the entire image to furnish new features that pertain to the scene's overview (an exemplification of this approach is provided by the Scenes Classification network). Conversely, the second approach aims to augment the VQA's capabilities to address questions pertaining to specific domains beyond its intrinsic capacity. In both scenarios, a register file houses the list of all registered modules, which the system accesses for reference. Each module must adhere to a predefined interface for implementation and furnish access to a class with a designated name. Specifically, image feature modules must provide a method that, when presented with an image, returns a class representative of the features extracted by the module (such as a scene descriptor for our Scenes Classification Network). On the other hand, VQA support modules must implement a method to determine if the module can address a given question. If affirmative, the module must also furnish a method to generate an answer when presented with an image.

6. Conclusions

In this study, we introduce a novel image captioning system designed to facilitate interactive exploration for visually impaired individuals. We present a framework that combines visual relation detection with visual question answering, enabling users to focus on specific points of interest within an image interactively. Our findings demonstrate that even with rudimentary images, this system can effectively discern the image's key features and cater to user inquiries, thereby satisfying user curiosity. Moreover, we showcase the system's versatility through the integration of a knowledge base, enabling it to respond to informed inquiries and incorporate additional feature extraction capabilities through modularity. We emphasize the significance of modularity within our system, exemplifying its functionality through networks dedicated to dog breed classification and scene classification. These examples elucidate how the system can be expanded to address specific requirements, underscoring modularity's pivotal role. Looking ahead, future iterations of the system aim to broaden its capabilities by incorporating additional information extraction methods from images. Specifically, we plan to implement models for clothing classification and genre discrimination, enhancing the system's versatility and utility. We envision that this interactive approach to image captioning will catalyze further exploration in the field, encouraging researchers to delve into dynamic image exploration methods that mirror humans' innate propensity to navigate scenes and focus on preferred details.

Acknowledgments

This work has been developed at is.Lab() Intelligent Systems Laboratory at the Department of Computer, Control, and Management Engineering, Sapienza University of Rome (https:// islab.diag.uniroma1.it). Their preliminary studies have been done with the support of the students Angrisani and Lo Befaro. The work has also been partially supported from Italian Ministerial grant PRIN 2022 "ISIDE: Intelligent

Systems for Infrastructural Diagnosis in smart-concretE", n. 2022S88WAY - CUP B53D2301318, and by the Age-It: Ageing Well in an ageing society project, task 9.4.1 work package 4 spoke 9, within topic 8 extended partnership 8, under the National Recovery and Resilience Plan (PNRR), Mission 4 Component 2 Investment 1.3—Call for tender No. 1557 of 11/10/2022 of Italian Ministry of University and Research funded by the European Union—NextGenerationEU, CUP B53C22004090006.

References

[1] F. Bonanno, G. Capizzi, A. Gagliano, C. Napoli, Optimal management of various renewable energy sources by a new forecasting method, 2012, pp. 934 – 940. doi:10.1109/SPEEDAM.2012.6264603.

[2] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, A multithread nested neural network architecture to model surface plasmon polaritons propagation, Micromachines 7 (2016). doi:10.3390/mi7070110.

[3] G. Capizzi, C. Napoli, S. Russo, M. Woźniak, Lessening stress and anxiety-related behaviors by means of ai-driven drones for aromatherapy, volume 2594, 2020, pp. 7 – 12.

[4] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection 14126 LNAI (2023) 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[5] I. E. Tibermacine, A. Tibermacine, W. Guettala, C. Napoli, S. Russo, Enhancing sentiment analysis on seed-iv dataset with vision transformers: A comparative study, 2023, pp. 238 – 246. doi:10.1145/3638985.3639024.

[6] N. N. Dat, V. Ponzi, S. Russo, F. Vincelli, Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches, volume 3118, 2021, pp. 51 – 63.

[7] V. Ponzi, S. Russo, A. Wajda, R. Brociek, C. Napoli, Analysis pre and post covid-19 pandemic rorschach test data of using em algorithms and gmm models, volume 3360, 2022, pp. 55 – 63.

[8] E. Iacobelli, V. Ponzi, S. Russo, C. Napoli, Eye-tracking system with low-end hardware: Development and evaluation, Information (Switzerland) 14 (2023). doi:10.3390/info14120644.

[9] F. Fiani, S. Russo, C. Napoli, An advanced solution based on machine learning for remote emdr therapy, Technologies 11 (2023). doi:10.3390/technologies11060172.

[10] L. Cewu, K. Ranjay, B. Michael, F.-F. Li, Visual relationship detection with language priors, European Conference on Computer Vision (2016).

[11] R. Joseph, D. Santosh, G. Ross, F. Ali, You only look once: Unified, real-time object detection, Conference on Computer Vision (2016).

[12] J. Jaewon, P. Jongyoul, Visual relationship detection with language priors and softmax, European Conference on Computer Vision (2019).

[13] J. Lu, J. Yang, D. Batra, D. Parikh, Hierarchical question-image co-attention for visual question answering 29 (2016). URL: https://proceedings.neurips.cc/paper/2016/file/9dcb88e0137649590b755372b040afad-Paper.pdf.

[14] W. Qi, T. Damien, W. Peng, S. Chunhua, D. Anthony, v. d. H. Anton, Visual question answering: A survey of methods and datasets, Computer Vision and Image Understanding (2017).

[15] R. Delia, D. Lorand, B. Fortuna, M. Grobelnik, D. Mladení, Triplet extraction from sentences (2007).

[16] E. Charniak, M. Johnson, Coarse-to-fine n-best parsing and MaxEnt discriminative reranking (2005) 173–180. URL: https://www.aclweb.org/anthology/P05-1022. doi:10.3115/1219840.1219862.

[17] M. Diego, F. Anton, T. Ivan, A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling, Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada (2017) 411–420.

[18] Z. Yang, X. He, J. Gao, L. Deng, A. Smola, Stacked attention networks for image question answering (2016) 21–29. doi:10.1109/CVPR.2016.10.

[19] G. Danna, L. Qing, L. Chi, Z. Yinan, G. Anhong, J. Abigale, L. Stangl, P. Jeffrey, Vizwiz-priv: A dataset for recognizing the presence and purpose of private visual information in images taken by blind people (2019). URL: https://www.ischool.utexas.edu/~dannag/publications/CVPR2019_VizWiz-Priv.pdf.

[20] R. Mengye, K. Ryan, Z. Richard, Exploring models and data for image question answering (2015).

[21] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories 2 (2006) 2169–2178. doi:10.1109/CVPR.2006.68.