



**BRNO UNIVERSITY OF TECHNOLOGY**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FACULTY OF INFORMATION TECHNOLOGY**

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

**VISUAL QUESTION ANSWERING**

SYSTÉM PRO ODPOVÍDÁNÍ NA OTÁZKY S VYUŽITÍM OBRAZU

**BACHELOR'S THESIS**

BAKALÁŘSKÁ PRÁCE

**AUTHOR**

AUTOR PRÁCE

**SUPERVISOR**

VEDOUCÍ PRÁCE

**PAVEL KOCUREK**

**Ing. MARTIN FAJČÍK**

**BRNO 2021**

# Bachelor's Thesis Specification



Student: **Kocurek Pavel**  
Programme: Information Technology  
Title: **Visual Question Answering**  
Category: Speech and Natural Language Processing

## Assignment:

1. Describe the problem of visual question answering. Compare various problem perspectives studied in this research area.
2. Research the current state-of-the-art methods for a given problem.
3. Find an application area where such a visual question answering application might be useful.
4. Describe available datasets for the problem.
5. Choose and describe a suitable method.
6. Design and implement the application useful in the selected area. The application will be based on the selected method.
7. Evaluate the method.

## Recommended literature:

- Teney, D., Anderson, P., He, X. and van den Hengel, A., 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4223-4232).
- Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T. and Rohrbach, M., 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*.
- Hudson, D.A. and Manning, C.D., 2019. Learning by Abstraction: The Neural State Machine. *arXiv preprint arXiv:1907.03950*.

## Requirements for the first semester:

- Complete items 1 to 4 of the assignment.

Detailed formal requirements can be found at <https://www.fit.vut.cz/study/theses/>

Supervisor: **Fajčík Martin, Ing.**  
Head of Department: Černocký Jan, doc. Dr. Ing.  
Beginning of work: November 1, 2020  
Submission deadline: May 12, 2021  
Approval date: April 1, 2021

## Abstract

Visual Question Answering (VQA) is a system where an image and a question are used as input and the output is an answer. Despite many research advances, unlike image captioning, VQA is rarely used in practice. This work aims to narrow the gap between research and practice. To examine the possibility of using VQA by blind and visually impaired people, this thesis proposes a demonstrative VQA application and then, a smartphone application. The study with 20 participants from the community was conducted. Firstly, the participants received an application for two weeks. Then, each of them was asked to fill out the questionnaire. 80% of respondents rated the accuracy of VQA application as sufficient or better and most of them would appreciate it if their image captioning application also supported VQA. Following this discovery, this work tries to establish the link between image captioning and VQA. In particular, the work studies the informativeness provided by both systems in different scenarios. It collects a novel dataset of 111 images with manually annotated captions and diverse scenes. An experiment comparing obtained knowledge showed a success rate of 69.9% and 46.2% for VQA and image captioning, respectively. In another experiment 70.9% of the time, participants were able to select the correct caption based on VQA. The results suggest that VQA outperforms image captioning regarding image details, therefore should be used in practice more often.

## Abstrakt

Visual Question Answering (VQA) je systém, kde je vstupem obrázek s otázkou a výstupem je odpověď. Navzdory mnoha pokrokům ve výzkumu se VQA, na rozdíl od počítačově generovaných popisů obrázků, v praxi používá jen zřídka. Cílem této práce je zúžit mezeru mezi výzkumem a praxí. Z tohoto důvodu byla kontaktována komunita zrakově postižených a byla jim nabídnuta demonstrativní aplikace VQA a následně byla vytvořena mobilní aplikace. Byla provedena studie s 20 účastníky z komunity. Nejprve účastníci zkoušeli demonstrativní aplikaci po dobu dvou týdnů a následně byli požádáni o vyplnění dotazníku. 80 % respondentů hodnotilo přesnost aplikace VQA jako dostatečnou nebo lepší a většina z nich by ocenila, kdyby jejich aplikace pro generování popisů podporovala také VQA. Po tomto zjištění práce porovná získané znalosti z VQA se znalostmi z popisů v různých scénářích. Byla vytvořena datová sada 111 obrázků různorodých scén s ručně anotovanými popisky. Experiment porovnávající získané znalosti ukázal úspěšnost 69,9 % pro VQA a 46,2 % pro popisy obrázků. V dalším experimentu v 70,9 % případů účastníci vybrali správný popis za pomoci VQA. Výsledky naznačují, že pomocí VQA je možné zjistit více znalostí o detailech obrázků než je to v případě generovaných popisů.

## Keywords

visual question answering, computer vision, natural language processing, question answering, image captioning, deep learning, questionnaire, rnn, lstm, bert, object detection

## Klíčová slova

odpovídání na otázky z obrazu, zpracování přirozeného jazyka, odpovídání na otázky, popisování obrázku, hluboké učení, dotazník, rnn, lstm, bert, detekce objektů

## Reference

KOCUREK, Pavel. *Visual Question Answering*. Brno, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Martin Fajčík

## Rozšířený abstrakt

Náplní této práce je Visual Question Answering (VQA) neboli systém pro odpovídání na otázky s využitím obrazu. Tento systém je tvořen kombinací počítačového vidění a zpracování přirozeného jazyka. Vstupem do systému je libovolný obrázek a otázka k tomuto obrázku a výstupem je odpověď na danou otázku v přirozeném jazyce. Každý rok dochází k mnohým pokrokům ve výzkumu VQA. Na druhou stranu, reálné využití ve veřejně dostupných aplikacích není téměř žádné. Tato práce zkoumá možné využití VQA v životech nevidomých a jinak zrakově postižených. Úkony jako například výběr oblečení nebo orientace ve městě jsou pro mnohé vidomé jedince bezproblémové. Pro nevidomé se ale mnohdy jedná o nepředstavitelné obtížnosti. Na rozdíl od VQA, systém pro generování popisů obrázku neboli image captioning (IC) je využíván výrazně častěji. Z 10 testovaných aplikací často využívaných slepými jich 6 poskytovalo IC ale žádná z nich neposkytovala VQA. Kromě zaměření na nevidomé se tato práce snaží porovnat VQA a IC a zjistit, proč se VQA v praxi využívá podstatně méně často.

První kapitola se zabývá vysvětlením základních principů, nutných k pochopení fungování celku. Jedná se nejprve o zpracování přirozeného jazyka, dále pak rekurentní neuronové sítě a základní koncepty počítačového vidění. Následuje popis způsobu, jakým nevidomí využívají technologie. Přestože existují speciální zařízení jako například Braillovy displeje, většina nevidomých jej nevyužívá z důvodů, že si je nemohou finančně dovolit, nebo protože neovládají Braillovo písmo. Tito lidé jsou tedy při konzumaci informací u počítače nebo chytrého telefonu odkázáni na čtečku obrazovky, případně klávesnici nebo dotykovou obrazovku. V případě telefonu často využívají aplikace pro popis obrázků. Poslední část této kapitoly se zabývá statistickou mírou Fleissova kappa, sloužící pro výpočet shody mezi více hodnotiteli.

Další kapitola se již věnuje konkrétně přístupům a problémům VQA. Nejprve jsou popsány nejdůležitější datové sady a jejich způsoby vyhodnocení. Poté jsou vysvětleny metody posledních let, jež dosahují nejlepších výsledků.

Následující kapitola se již zabývá možným využitím pro VQA zejména tedy použití nevidomými a zkoumáním, zda by pro ně bylo VQA užitečné. Nejprve jsou porovnány mobilní aplikace, jež jsou nejčastěji využívány slepými. Pro ověření, zda by VQA mohlo nevidomým nebo jinak zrakově postiženým lidem pomoci byla vytvořena demonstrační aplikace. Tuto aplikaci lidé vyzkoušeli a následně mohli vyplnit dotazník. Dotazovaní byli různého zrakového postižení a více než polovina z nich je starších 40 let. Většina respondentů by ocenila, kdyby jejich aplikace pro popis obrázků umožňovala také možnost odpovědět na otázky z obrazu, tedy VQA. Nejčastější možné využití dle tázaných je orientace v prostoru (60%), rozpoznávání předmětů a jejich lokalizace (45%), výběr oblečení (35%), případně vaření nebo online nakupování. Celkové hodnocení aplikace bylo kladné, nicméně výsledky by mohly být ovlivněny skutečností, že pokud někdo nebyl spokojen s používáním, nemusel dotazník vyplnit vůbec.

Další kapitola se již zabývá srovnáním VQA a IC. Za tímto účelem byla vytvořena datová sada obsahující 111 obrázků. Ke každému obrázku byly ručně vytvořeny 3 popisy různými lidmi a jedna otázka. Otázka byla vytvořena tak, aby její tvůrce před vytvořením neviděl popisy. Nejprve byly použity dvě různé VQA metody, kdy byly jejich vstupem obrázek a otázka z vlastní datové sady. Výsledky byly vyhodnoceny vlastní metrikou a pro další experimenty byla vybrána metoda, jež dosáhla vyššího skóre. Dále byl pro každý obrázek vygenerován popis systémem pro generování popisů. Následující experiment spočíval ve srovnání získaných znalostí z popisu a z VQA. Účastník obdržel generovaný, nebo člověkem vytvořený popis a otázku. Jeho úkolem bylo odpovědět za předpokladu, že z



daného popisu je možné zjistit informaci nutnou k zodpovězení otázky. Tento proces byl opakován pro všechny otázky. Získané výsledky byly následně porovnány s výsledky získané VQA modelem při jejímž vyhodnocení. VQA model zde dosáhl přesnosti 69,9%, zatímco člověkem vytvořené popisy dosáhli 58,1% a generované popisy dokonce 46,2%. Na základě těchto výsledků je možné usoudit, že VQA dokáže lépe zachytit specifické detaily obrázku v porovnání s IC. Cílem následujícího experimentu bylo vyhodnotit, zda je pomocí VQA možné zjistit informace o obrázku, aniž by bylo možné daný obrázek vidět. Účastník obdržel 5 různých popisů, kdy pouze jeden z nich byl správný a ostatní byly náhodně vybrány ze zbylých popisů v datové sadě. Účastník se mohl pomocí VQA zeptat na 1-3 otázky a na základě zjištěných informací vybral jeden z popisů. Tento experiment byl proveden třemi účastníky, kteří v průměru dosáhli přesnosti 70,81%. Tyto výsledky naznačují, že by VQA mohlo být vhodné pro nevidomé, jelikož ve většině případů dokáže pomocí několika otázek zjistit obsah obrázku.

Všechny experimenty v této práci byly vyhodnoceny na základě vlastních metrik. Pro porovnávání popisů by mohlo být využito například BLEU, nicméně tato metrika trpí zásadními nedostatky. Nebere v potaz smysl, stavbu věty, ani synonyma. V případě VQA odpovědí by bylo možné porovnat s několika možnými správnými odpověďmi, avšak zde nastává stejný problém v případě synonym. Použité metriky mohou působit subjektivně, nicméně pro datovou sadu této velikosti je upřednostněn lidský úsudek.

Generované popisy jsou na obrázku schopné zachytit některé objekty, případně jejich vztahy. Nicméně, tyto popisy mají problém zachytit specifické detaily. Tento problém by mohla řešit kombinace generovaných popisů a VQA, jenž se dokáže zaměřit na konkrétní detaily obrázku. Na druhou stranu tato kombinace zvyšuje komplexitu. Popisy nevyžadují žádný další vstup mimo obrázek, zatímco pro VQA je nutné vytvořit otázku. K získání užitečných znalostí je nutné tuto otázku správně formulovat.

# Visual Question Answering

## Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Martin Fajčík. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....  
Pavel Kocurek  
May 9, 2021

## Acknowledgements

I would like to thank my supervisor Ing. Martin Fajčík for his guidance, enthusiasm, patience and willingness to answer all my questions or help with any issues along the way. Thank You.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Prerequisite Concepts</b>	<b>4</b>
2.1	Natural Language Processing . . . . .	4
2.2	Recurrent Neural Network . . . . .	5
2.2.1	Transformer . . . . .	7
2.2.2	BERT . . . . .	9
2.3	Computer Vision . . . . .	11
2.3.1	Object detection . . . . .	12
2.3.2	Image Captioning . . . . .	16
2.4	Technology for the Blind . . . . .	17
2.5	Fleiss' Kappa . . . . .	19
<b>3</b>	<b>Visual Question Answering</b>	<b>20</b>
3.1	Datasets and Evaluation . . . . .	22
3.2	Methods . . . . .	27
<b>4</b>	<b>Usage Scenarios</b>	<b>31</b>
4.1	Existing Applications . . . . .	31
4.2	Demonstrative Application . . . . .	33
4.3	Testing by Blind and Visually Impaired . . . . .	35
4.4	Smartphone Application . . . . .	38
<b>5</b>	<b>VQA versus Image Captioning</b>	<b>41</b>
5.1	Dataset Collection . . . . .	41
5.2	Visual Question Answering . . . . .	43
5.3	Image Captioning . . . . .	44
5.4	Comparison of Obtained Knowledge between VQA and Image Captioning . . . . .	46
5.5	Reasoning over Images with VQA . . . . .	48
5.6	Summarizing Discussion . . . . .	50
<b>6</b>	<b>Discussion and Conclusion</b>	<b>51</b>
	<b>Bibliography</b>	<b>52</b>

# Chapter 1

## Introduction

Although human understanding of the image for machines is complex to understand, the rapid increase in computing power makes it possible to create systems and problems that were unthinkable not so many years ago. Visual Question Answering (VQA) is one of many tasks, that took advantage of that. The VQA takes an image and an arbitrary textual question about that image as input and its output is a generated answer. An example could be an image of a parking lot with a question of how many available parking places are there, or any other image and question such as Figure 1.1.

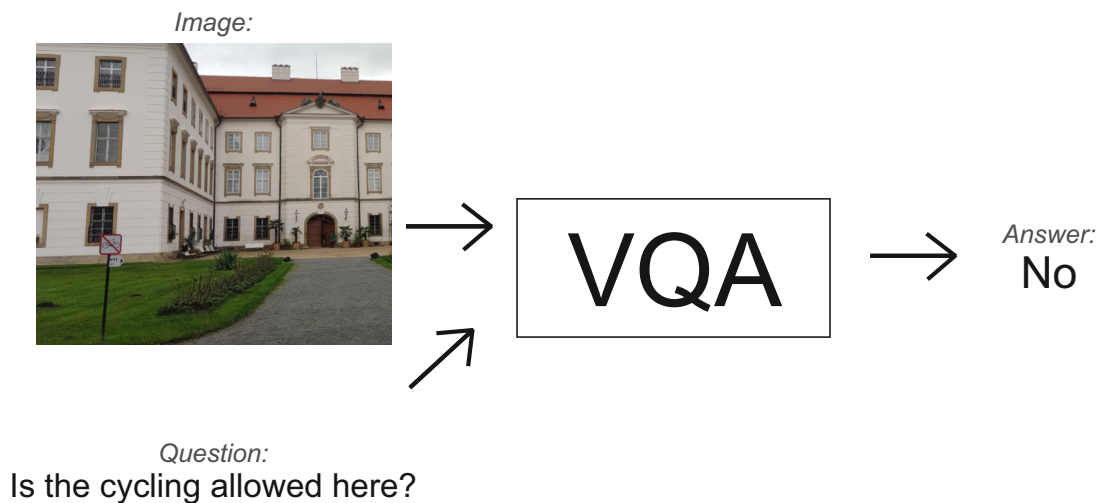


Figure 1.1: An example where the VQA system determines the answer to a question based on an image.

The aim of this thesis is to narrow the gap between research and practice. Most of the time, it does not take more than a few months for a VQA method to surpass the previously best-performing method. VQA, its methods and datasets are studied for many years [2, 57, 63, 30], yet there is a lack of publicly available applications.

This work founds that in practice, there is a common belief that image captioning (IC) — a closely related task of generating a description for an image scene — provides enough information to understand the image contents. But does it? To answer this question I turn towards the community which is in research papers often targeted as an example of user-base that could benefit from VQA and IC. Of the 10 mobile application from Google Play (Android) and App Store (iOS) with more than 50K downloads, it was found that 6 of them provide IC but none offer VQA.

This thesis tries to determine whether VQA could also benefit the blind and visually impaired (BVI) community in practice. First, a VQA demonstrative application was created, then used by BVI. The knowledge was gathered using a questionnaire, and a smartphone application was created. Then, a novel dataset of images, captions and questions was manually collected. Next, another set of captions was generated by using an image captioning model. Followed by experiments with VQA and a novel dataset were performed. These experiments were manually evaluated, and results were discussed. The first tries to determine if image captioning could be as informative as VQA. An experiment was designed where participants had to answer questions based purely on image captions. The same questions, this time with images were fed to VQA and then the results were compared. The aim of the second experiment was to find out, how well can user obtain knowledge about image with VQA without seeing the actual image.

In Chapter 2 are presented key concepts needed to understand the inner workings of VQA. Natural language processing (NLP) combines linguistics, computer science and machine learning. Recurrent neural network or a modification such as Long short term memory are the essential elements of deep learning. Transformer and BERT are more advanced deep learning models used for the NLP. Since VQA combines image and text processing it intervenes with areas of NLP, and also Computer vision. The next section covers technologies used by BVI and the last section explains a statistical measure used later in experiments. Following Chapter 3 describes VQA and its problem perspectives, the most important datasets with their evaluations and also the state-of-the-art methods of recent years. The next Chapter 4 examines the possible use case of VQA for the BVI community. At first, similar applications are studied, then a demonstrative application is created. BVI people try the application and afterwards a questionnaire is used to determine if VQA could be helpful and then, a smartphone application is created. Chapter 5 consists of a collection of the novel dataset, which is used for experiments about VQA and IC. Then, these experiments are presented and the results are discussed in the last section. Finally, the work and its observations are summarized in Chapter 6.

## Chapter 2

# Prerequisite Concepts

To understand the rest of this thesis, it is necessary to cover few key concepts. The first of all is covered the Natural Language Processing (Section 2.1). Follows an introduction to Recurrent Neural Network (Section 2.2), Transformers and BERT. Another section covers Computer Vision (Section 2.3), its history and object detection based on Convolutional Neural Networks. Next is explained, how do blind people use technologies (Section 2.4). Finally is covered statistical measure Fleiss' Kappa (Section 2.5).

### 2.1 Natural Language Processing

Natural language processing (NLP) refers to the automatic computational processing of human languages. This encapsulates algorithms that take human-produced text as input and those, that produce natural-looking text as output. NLP techniques were dominated by linear modelling approaches to supervised learning, trained over very high dimensional yet very sparse feature vectors, for more than a decade. This paradigm began to shift around 2013 with word2vec [43] when neural network models over dense inputs began to gain success. Human language is immensely variable and ambiguous and also ever-changing and evolving. Although humans are capable of understanding and producing data in language, for computers, it is rather challenging. Machine learning methods shine at problems, where even though a good set of rules is hard to establish if the expected output for a given input is simple enough. Language is symbolic, discrete, and compositional. The basic components of language are morphemes. These characters are composed into words, and words form phrases and sentences. The meaning of a word can differ if it is used on its own, are in a phrase. Thus, to be able to interpret the text, it is necessary to understand characters and words and also sentences and even larger spans of text [15].

#### Deep Learning in NLP

The scientific discipline studies understanding of written and spoken language from a computational perspective. To be able to perform any linguistic task, the machines are required to comprehend the structure of a language. In recent years huge leap forward in this area was done by many researchers [6, 23, 49], which finally focused on language as a whole, rather than just optimizing for a specific task. The language can be broken down into multiple areas such as, morphology, language modelling, semantics and parsing [15].

Morphology studies the words and how are they made. It considers not only the roots of words, prefixes, and suffixes but also compounds, plurality, gender and others. Language modelling establishes what are the interactions from word to word and which should follow which. The area of semantics studies the meaning of individual words, what are their relations to each other and the context, they appear. Finally, the parsing examines which words modify others and the overall structure of a sentence [46]. A crucial component of neural networks for language is the embedding layer. It provides the interface for mapping words or other discrete symbols to continuous vectors in a low dimensional space. Upon these vectors can be performed mathematical operations. Distance between specific vectors can represent relations between corresponding words. The representation of words is learned during the learning process [15].

## 2.2 Recurrent Neural Network

In a feedforward network, information is transferred in only one direction from input to output, one layer at a time. In Recurrent Neural Network (RNN) the output of a layer is added to the next input and fed back into the same layer . Unlike feedforward networks, RNN can receive as input a sequence of values and can also produce a sequence of values as output. Typically, RNN is difficult to train. Due to backpropagation, there is a problem of exploding and vanishing gradient [48], a cause of significant decay of information through time. There are many approaches to deal with this issue, such as using gradient clipping [71], skip connections<sup>1</sup> or rectified linear activation function<sup>2</sup>.

### Long Short Term Memory

Long Short Term Memory [22] (LSTM) addresses the problem of vanishing gradient by introducing the long-term memory called the **cell state** denoted by  $C_t$  and represented by a horizontal line in Figure 2.1. LSTM does have the ability to remove or add information to the **cell state**, carefully regulated by structures called **gates**. The input vector  $[h_{t-1}, x_t]$  is an input to LSTM at time step  $t$ . The output vector at time step  $t$  is denoted as  $h_t$ . Concatenation in equations is represented as “,” symbol and “ $\circ$ ” stands for Hadamard product. This subsection is inspired by Understanding LSTM Networks [45].

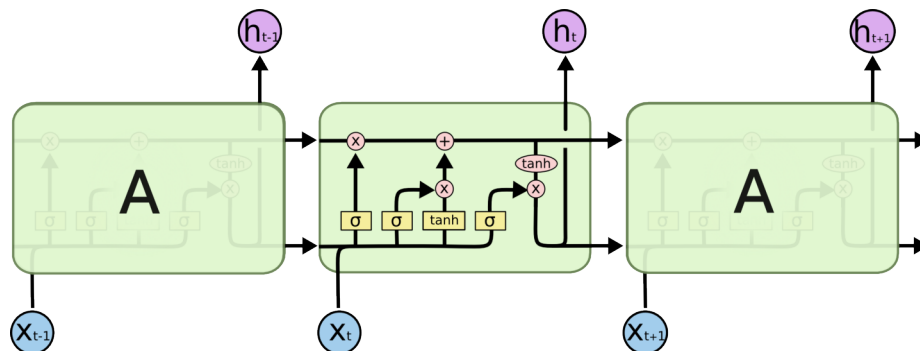


Figure 2.1: Long Short Term Memory cells (Source [45]).

<sup>1</sup><https://theaisummer.com/skip-connections/>

<sup>2</sup>[https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

Each unit contains **hidden state**, three **gates** and a **cell state**. Gate layers use the sigmoid activation (Figure 2.2) since it outputs a value between 0 and 1, it can either let no flow or complete flow of information throughout the gates.

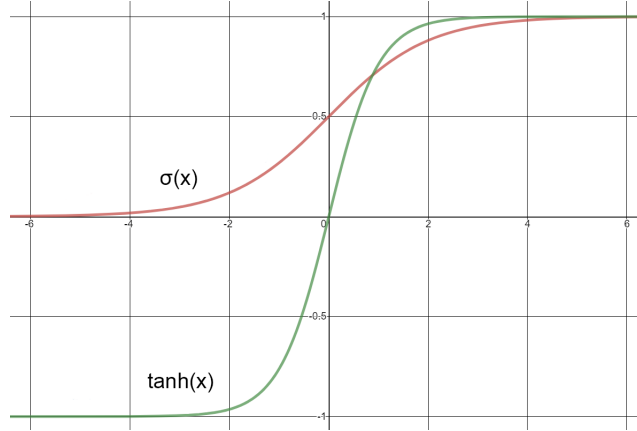


Figure 2.2: Sigmoid and Tanh activation functions (  $\sigma(x) = \frac{1}{1+e^{-x}}$ ,  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  ).

The first layer (Equation 2.1) **Forget gate** combines information from the previous hidden state and the current input. Based on the output of the sigmoid function, the values closer to 0 are forgot and closer to 1 are kept.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.1)$$

The next layer, **Input gate** consists of two parts. A hyperbolic tangent is used to regulate the network by ensuring that the values stay between -1 and 1. The sigmoid layer decides what new information should be stored. These values are multiplied and then added to the Hadamard product of the forget gate and previous cell state.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (2.3)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \quad (2.4)$$

The last layer, the **Output gate** besides returning **cell state** feeds the state to *tanh* function. The result is then multiplied with a **hidden state** adjusted with *sigmoid* function. The output of these operations is the new **hidden state**.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t \circ \tanh(C_t) \quad (2.6)$$

One of the popular modifications of LSTM is the Gated Recurrent Unit [3]. The main difference is using two **gates** rather than three. The **hidden state** is merged with the **cell state** and the **forget gate** and **input gate** are combined into the **update gate**. The GRU requires fewer training parameters, uses less memory, therefore, the training is more efficient. Mostly both architectures yield comparable performance and tuning hyper-parameters could be more beneficial than choosing architecture. Research [4] has shown no concrete conclusion on which of the two gating units is better.



## 2.2.1 Transformer

Recurrent models use previous states as input for computation of the current state, which means that computation is sequential. This becomes an issue for longer sequences, as it requires extensive memory usage, which is an even bigger issue for Transformer. This section is inspired by [59]. The Transformer is the model, that uses a self-attention mechanism, which allows relating different positions of the input sequence to compute its representations. The Transformer is based on encoder-decoder architecture [58] combining self-attention with fully connected layers.

### Model Architecture

The architecture proposed in the original paper (Figure 2.3) is made of  $N = 6$  identical layers of encoders on the left and decoders on the right side of the figure. The input and output embedding is added to the positional encoding to determine a position in the sequence as the self-attention is position invariant, and then fed into the encoder, decoder respectively. The encoder maps a sequence of symbol representations  $(x_1, \dots, x_n)$  to a continuous sequence  $z = (z_1, \dots, z_n)$ . Based on  $z$  the decoder generates at each time step one symbol from a sequence of output symbols  $(y_1, \dots, y_m)$ .

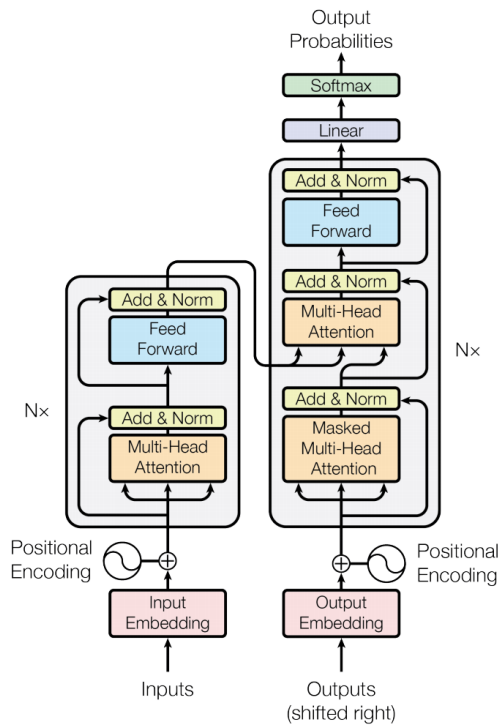


Figure 2.3: The architecture of Transformer with  $N$  layers of encoder on the left and decoder on the right (Source [59]).

The encoder consists of two sub-layers, each with its own residual connection [20] to counteract the problem of exploding and vanishing gradient, both followed by layer normalization [66]. Residual connections allow gradients to flow through a network directly rather than passing through a non-linear activation function. The first is multi-head self-attention and the second fully connected feedforward network. In addition to encoder layers, the

decoder contains one more layer, the masked multi-head attention. By modifying the self-attention in combination with offsetting the output embeddings by one position is prevented from attending subsequent positions. The reason is to prevent leftward information flow in the decoder to preserve the auto-regressive property.

### Self-Attention

Self-attention relates different positions of a single sequence to compute a representation of the sequence. It enables to find correlations between different tokens of the input. The embedding for each word is used to create query  $Q$ , key  $K$  and value  $V$  matrices. This is achieved by multiplying an embedding matrix  $X$  with weight matrices  $W_Q, W_K$  and  $W_V \in \mathbb{R}^{d_{model} \times d_k}$ , where  $d_{model}$  is the output vector size. Key and query matrices share dimensions  $d_k$ . A softmax function is applied to get the final attention weights as a probability distribution. The attention is computed using Equation 2.7.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.7)$$

The comparison of different layer types is in Table 2.1. The idea of self-attention is expanded into multi-head attention. This way the model is able to better capture positional information. The output vector size is divided by the number of heads. In original paper authors use  $d_{model} = 512$  with  $h = 8$  heads. The heads are concatenated and transformed using a square weight matrix  $W^O \in \mathbb{R}^{hd_k \times d_{model}}$  (Equation 2.8).

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (2.8)$$

### Position-wise Feedforward Networks

Each of the encoder and decoder contains a fully connected feedforward neural network composed of two linear transformations and a ReLU activation (defined as  $\max(0, x)$ ).

$$\text{FFN}(x) = \max(0, W_1x + b_1)W_2 + b_2 \quad (2.9)$$

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Table 2.1: Comparison of per-layer computational complexity, minimum number of sequential operations and maximum path lengths across time between any two positions in the network for different types of layers.  $n$  is the sequence length,  $d$  is the dimension,  $k$  is the kernel size of convolutions and  $r$  is the size of the neighborhood (Source [59]).

## 2.2.2 BERT

Bidirectional Encoder Representations from Transformers [6] (BERT) is a machine learning technique for natural language processing. Architecture is a multi-layer bidirectional Transformer encoder based on [59]. Directional models process input sequentially. BERT on the other hand handles the entire input sequence at once, thus it is able to learn the context of a word based on all of its surroundings. The training consists of two phases. First, the model is pre-trained to learn the language structure and in the second phase fine-tuned<sup>3</sup> for a specific task. Training is visualised in Figure 2.4. In the original paper were reported two types of architecture. The first  $BERT_{BASE}$  is made of 12 layers, hidden size of 768, 12 self-attention heads and 110M total parameters. on the other hand, there is  $BERT_{LARGE}$  with 24 layers, 1024 hidden size, 16 self-attention heads and 340M total parameters.

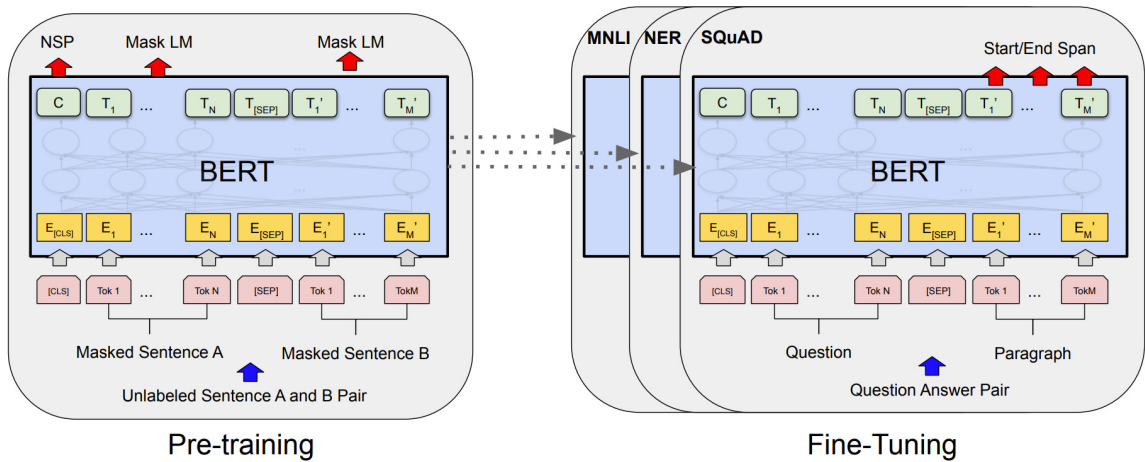


Figure 2.4: Visualization of two phases of training BERT model. On the left is learning of language structure and on the right is fine-tuning for various tasks. [CLS] is a special symbol added at the start of each input and [SEP] is a separator token (Source [6]).

### Pre-training BERT

The BERT is trained simultaneously on two unsupervised tasks, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). A combination of those two allows BERT to get decent knowledge of language structure. For the MLM BERT takes as input sentences with randomly masked 15% of all tokens, and the model tries to predict the masked words. The chosen tokens are not always masked. Since masked tokens do not appear during fine-tuning, chosen token is masked 80% of the time, replaced with random token 10% of the time and is not changed 10% of the time. It helps BERT understand the bi-directional context within a sentence. In the case of NSP, the BERT takes as input two sentences and it determines if the second sentence actually follows the first. Implementation is that in 50% of cases B is the actual next sentence that follows A and in 50% of cases it is a random sentence from corpus [6].

<sup>3</sup>Fine-tuning is a process after training of adjusting parameters to achieve the best possible results executed on a small set of data.

Different BERT variants such as ALBERT [34] replace NSP with a different task, such as Sentence Order Prediction (SOP). The authors of ALBERT claim that NSP conflates topic prediction and coherence prediction. Although NSP learns whether the two sentences belong to the same topic, determining if the sentences are grammatically coherent is a much harder task. The SOP allows the model to learn finer-grained distinctions about the coherence properties. For positive examples, the SOP loss uses two consecutive segments from the same document the same way as BERT. On the other hand, negative examples are used in the same consecutive segments with their order being swapped.

### Input Embedding

The input is a concatenation of two sentences with randomly chosen tokens being masked. BERT uses a WordPiece [64] vocabulary with a fixed size of roughly 30K tokens. A word that does not occur in a vocabulary is split into smaller and subwords and characters to create a token. A sequence is constructed from a pair of sentences separated with a token [SEP]. The first token for each sequence is a classification token [CLS]. The input embedding is denoted as  $E$ , the final hidden vector of the [CLS] token as  $C \in \mathbb{R}^H$  and the final hidden vector for the  $i^{\text{th}}$  input token as  $T_i \in \mathbb{R}^H$ . An initial embedding is constructed as a sum of three vectors (Figure 2.5) and then fed as input to BERT. In the training, the segment embedding represents segment number encoded into a vector. The position embedding is a position within the sentence encoded as a vector.

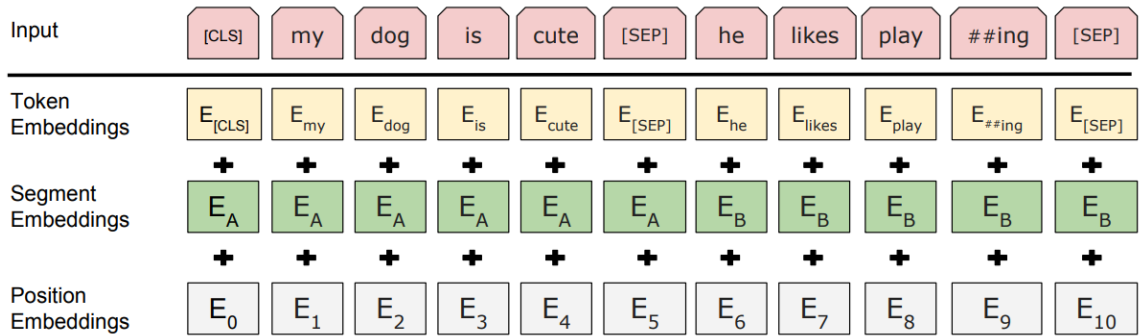


Figure 2.5: BERT input token embeddings (Source [6]).

### Fine-tuning BERT

In a fine-tuning example for question answering, the inputs are modified for a question followed by a text passage containing an answer. The next step is to perform supervised training using the QA dataset. It is only the output parameters that are learned from scratch. The rest of the model parameters is fine-tuned and as a result, the training is fast, and it can be done for various NLP tasks. What needs to be done is just replacing the output layers and then training with a specific dataset.

## 2.3 Computer Vision

Computer vision is the field of study focused on how computers perceive visual data such as digital images and videos. This interdisciplinary field simulates and automates elements of human vision systems using sensors, computers, and machine learning algorithms. This section is inspired by [32].

### History of Computer Vision

The research that preceded computer vision started more than 60 years ago. Authors of [24] studied the cat brain, which is similar to the human brain from a visual processing point of view. They found that visual processing begins with a simple structure of oriented edges, and as information moves along the path of visual processing, the brain creates the complexity of visual information until it can recognize a complex visual world.

More specific research followed after The Summer Vision Project (1966) [47], which was an attempt of MIT to use summer workers effectively in the construction of a significant part of a visual system. The book *Vision (the 1970s)* [42] explained how could be developed computer vision algorithms would enable computers to recognize the visual world. *Pictorial Structure* (1973) [9] studied ways to reduce the complex structure of an object to a set of simpler shapes and their geometric configurations. *Normalized Cut* (1997) [53] shows grouping pixels into meaningful areas using graph theory algorithm constructing fundamentals for image segmentation. The same year was released one of the main approaches to computer vision still widely used today, the convolutional neural network [36]. Two years later was published an important research considering image features [40] (Figure 2.6) and in 2001 authors of [62] were able to build near real-time face detector.

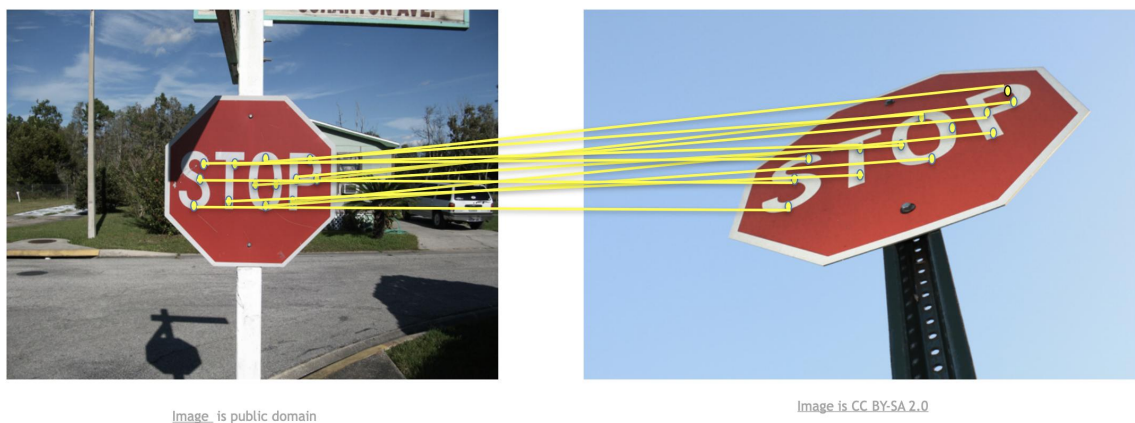


Figure 2.6: “SIFT” & Object Recognition, David Lowe, 1999 (Source [32]).

The early 21st century brings significant differences to this field. With the development of digital cameras and mobile phones, the quality and quantity of pictures is increasing. Numerous data sets [21, 5, 39] containing millions of images were created, and with those come benchmarks [7], making this field more competitive than ever before.

## Computer Vision Tasks

Apart from the visual question answering covered thoroughly in the next chapter, there are still many other challenging tasks to solve. These encapsulate image classification, face recognition, instance segmentation, semantic segmentation, image restoration, scene reconstruction and many more. Object detection and image captioning are crucial for this thesis.

### 2.3.1 Object detection

Whereas object localization is responsible for creating bounding boxes<sup>4</sup> for objects, image classification involves assigning labels and probabilities to those objects. Object detection is a combination of those previous two, thus creating bounding boxes and defining probabilities of labels corresponding to those boxes.

## Convolutional Neural Network

Convolutional Neural Network (CNN) is an alternative neural network most often used for image processing. The CNN is a sequence of convolutional, activation and pooling layers with the last layer being fully connected with SVM<sup>5</sup> or a softmax classifier. The convolutional layer is based on two-dimensional mathematical convolution defined for a two-dimensional image as Equation 2.10.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n), \quad (2.10)$$

where  $I$  is an input image,  $K$  stands for convolutional filter,  $i$  and  $j$  define location in an image and  $m, n$  represent size of convolutional filter [16]. An example of convolution is visualised in Figure 2.7 with two input channels and the same number of kernels. To each kernel is often added a certain bias. The result is a sum of convolutions of individual channels.

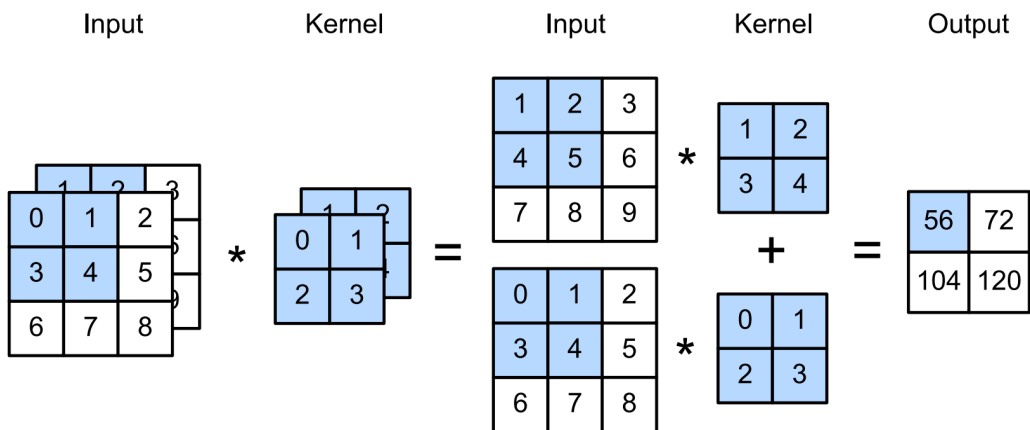


Figure 2.7: Two-dimensional convolution with kernel of size  $2 \times 2$  (Source [70]).

A pooling layer is often inserted between convolutional layers. The reasons are to reduce the computational cost and the dimensions of the feature maps by combining several

<sup>4</sup>Bounding box is a rectangular box used to determine the location of a target object in the image.

<sup>5</sup>[https://en.wikipedia.org/wiki/Support-vector\\_machine](https://en.wikipedia.org/wiki/Support-vector_machine)

values into one. The Pooling layer decreases the size by using operation such as maximum or average. The size of 3x3 is often used in practice because fields being too large would result in losing too much information.

The fully connected layer is a feed-forward neural network used to classify the data into various classes. The only difference between convolutional and fully connected layer is that many neurons in the convolutional volume share parameters and are connected only to a local region in the input [19].

The most common architectures are VGGNet (2014) [55] and ResNet (2015) [20]. VGG, as proposed in the paper, contains simple architecture of 16 layers and throughout the whole network is used convolution 3x3 and pooling 2x2. The model's depth is limited because of the vanishing and exploding gradient. These issues make deep convolution networks difficult to train.

Residual Network was proposed to mitigate the issue of vanishing gradient. The idea is to backpropagate through the identity function, by using vector addition. The shortcut connections perform identity mapping, and their outputs are added to the outputs of the stacked layers (Figure 2.8). Resnet also features heavy use of batch normalization [26] and is missing a fully connected layer at the output of the network.

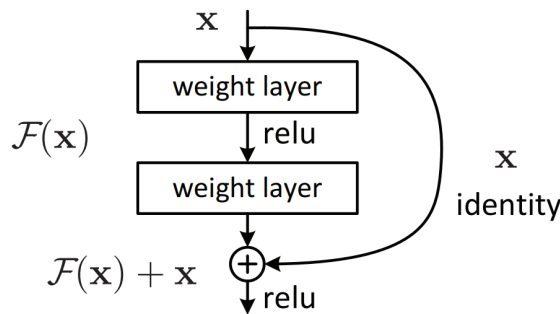


Figure 2.8: Residual block with an identity function to preserve the gradient (Source [20]).

## Faster R-CNN

Region-based convolutional neural networks (R-CNN) are a family of machine learning models for object detection. Faster R-CNN is a successor of R-CNN [13] and Fast R-CNN [12]. Faster R-CNN (2016) [51] is composed of a CNN followed by two trainable subnetworks. The region proposal network (RPN) proposes a set of rectangular objects with a membership score to a set of object classes/background. bounding boxes and the Fast R-CNN is used as a detector network. These networks share convolutional layers (Figure 2.9), thus accelerating the region proposal time from 2s to 10ms per image and also improving overall performance.

The region proposals are generated by sliding a small network over a convolutional feature map. The first step of RPN is the convolutional neural network with stride 16. This means that two points 16 pixels apart in the input image corresponding to two consecutive pixels in the output features. The RPN determines for every point of output whether



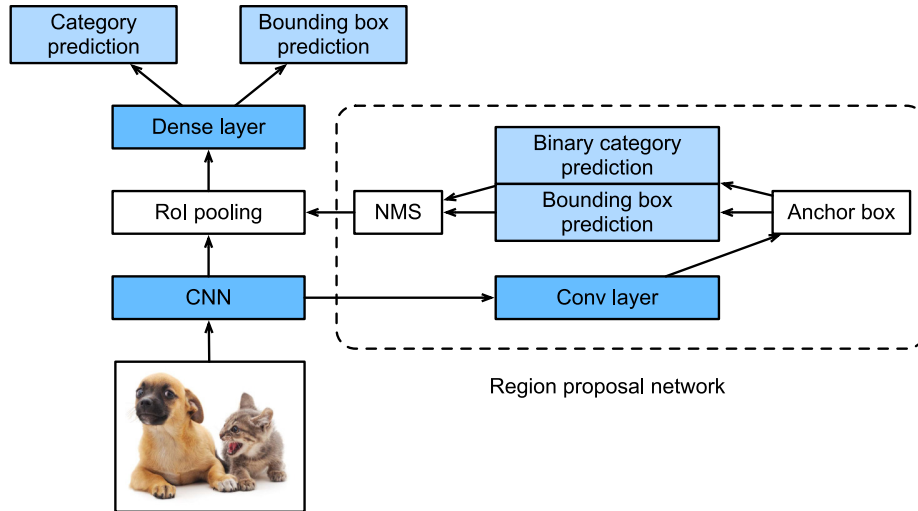


Figure 2.9: The architecture of Faster R-CNN is a Fast R-CNN with added Region proposal network (Source [70]).

an object is present at its corresponding location and estimate its size. The next step is to place a set of *anchors* (Figure 2.10) for each location on the output feature map. An *anchor* is located at the sliding window and is associated with a scale (3) and aspect ratio (3), thus for each sliding position  $k = 9$  anchors. Each sliding window is mapped to 256-dimensional feature, which is fed into the fully connected box-regression layer (*reg*) and a box-classification layer (*cls*).

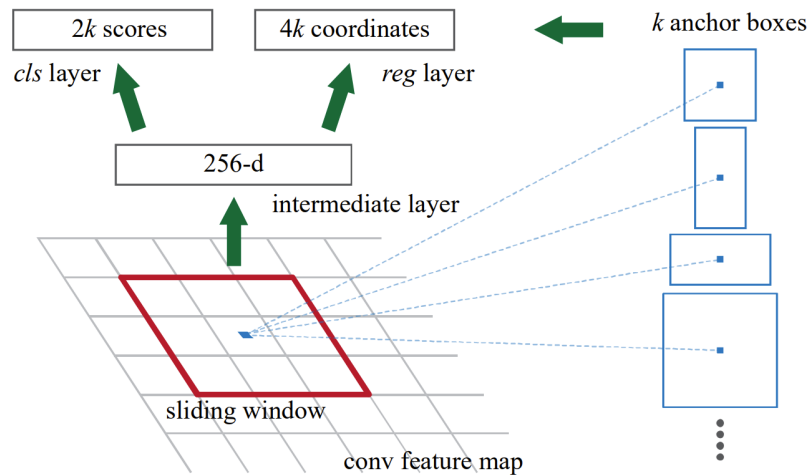


Figure 2.10: Region proposal network with number  $k$  of maximum possible proposals for each location. Scores represent estimate probability of object or not for each proposal (Source [51]).

The output feature map consists of about 20k *anchors* per image. These *anchors* indicate possible objects in various aspect ratios and sizes and are used for bounding box proposals. In the last step are discarded highly overlapping region proposals by using a non-maximum suppression (NMS) based on the Intersection-over-Union (IoU). Region



proposals with  $\text{IoU} \geq 0.7$  other than with the highest *cls* score are discarded.

For RPN training a binary class is assigned to each *anchor* based on their IoU overlap with ground-truth boxes. A positive label is assigned to anchor with the highest IoU or IoU overlap  $\geq 0.7$  with the ground truth box. on the other hand, a negative label is assigned to  $\text{IoU} \leq 0.3$  for all ground truth boxes. The multi-task loss is defined as a sum of classification loss and a bounding box regression loss  $L = L_{cls} + L_{box}$  (Equation 2.11).

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{box}} \sum_i p_i^* L_{box}(t_i, t_i^*), \quad (2.11)$$

where the classification loss  $L_{cls}$  is a cross-entropy over two classes (object or not), the output of *cls*  $\{p_i\}$  and *reg*  $\{t_i\}$  is normalized by mini-batch size  $N_{cls}$  (*i.e.*,  $N_{cls} = 256$ ) and a number of anchor locations  $N_{box}$  (*i.e.*,  $N_{box} \sim 2,400$ ) respectively and weighted by a parameter  $\lambda$ ,  $i$  is an anchor index,  $p_i$  is the predicted probability of  $i$  being an object. The ground-truth label  $p_i^*$  is 1 if anchor is positive,  $t_i$  is a vector representing predicted coordinates and  $t_i^*$  is that of the ground-truth box. Bounding box regression loss  $p_i^* L_{box}$  is activated only for positive anchors.

In Fast R-CNN the input is an image and a set of region proposals. The image is fed into the CNN to generate a convolutional feature map. For each region proposal, a region of interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map. RoI divides bounding boxes into a  $H \times W$  (*e.g.*,  $7 \times 7$ ) grid of sub-windows. For values in each sub-window is used max pooling. Pooling is applied to each feature map channel. Each RoI  $r$  is fed to two fully connected layers. The forward pass outputs a class posterior probability distribution  $p$  and a set of predicted bounding boxes. To  $r$  is assigned a detection confidence for each object class  $k$  using the estimated probability  $\Pr(\text{class} = k | r) \triangleq p_k$ . For all scored regions is applied NMS that rejects regions with IoU overlap larger than a learned threshold. The softmax classification layer assigns object classes and the bounding box regressor outputs coordinates for the bounding boxes (Figure 2.11). The computations of fully connected layers are accelerated by compressing with truncated SVD<sup>6</sup> [12].

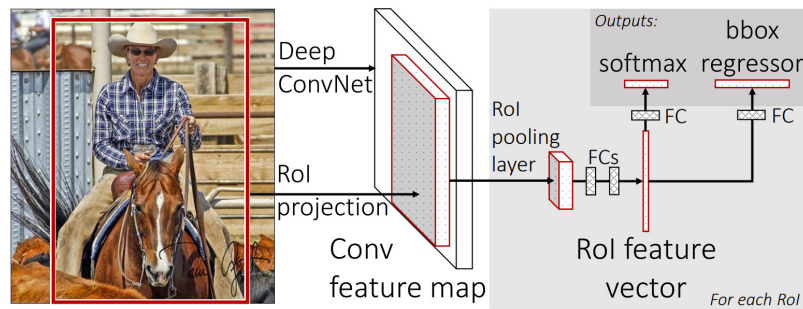


Figure 2.11: Fast R-CNN architecture (Source [12]).

<sup>6</sup>[https://en.wikipedia.org/wiki/Singular\\_value\\_decomposition](https://en.wikipedia.org/wiki/Singular_value_decomposition)

### 2.3.2 Image Captioning

Image Captioning (IC) is the computer science problem of generating a textual representation of an image (example in Figure 2.13). This requires the model to extract visual information from an image and understand language structure to be able to generate the corresponding caption. Despite a lot of efforts [67, 61, 8, 68], generated captions are still behind human captions. This problem is interesting because it has many important practical applications, such as enabling blind people to better understand their surroundings, but also because it deals with understanding of the image, which is a key part of computer vision.



The man at bat readies to swing at the pitch while the umpire looks on.



A large bus sitting next to a very tall building.



A horse carrying a large load of hay and two people sitting on it.



Bunk bed with a narrow shelf sitting underneath it.

Figure 2.12: An example of images and their generated captions from MS-COCO dataset (Source [39]).

Most of the existing approaches are either bottom-up and top-down. The bottom-up [8, 37] approach generates words describing multiple aspects of an image and then combine them into meaningful text. This approach suffers from problems such as generating too simple sentences or lacking the fluency of human writing. The top-down [61, 31] approach, on the other hand, starts with the context of an image and then uses words to describe it. [68] proposes another approach, the combination of the previous two through a semantic attention model.

One of the widely used models today [1] still relies on a combination of bottom-up and top-down attention mechanism. The bottom-up mechanism based on Faster R-CNN [51] combined with ResNet-101 [20] proposes salient image regions (Figure 2.13) with associated feature vectors. This combination allows selecting a relatively small amount of image bounding boxes from all possible configurations. The captioning model contains top-down

attention LSTM and language LSTM. The model’s objective is to minimize the cross-entropy loss, followed by Self-Critical Sequence Training [52]. Since [1] the salient image regions were the preferred approach by most researchers. The researchers such as authors of [28] suggest using grid features once again. By using grids, they were able to achieve comparable performance to salient image regions. Some of the benefits are simpler and faster computations and high recall since this approach covers the entire image rather than sparse regions.

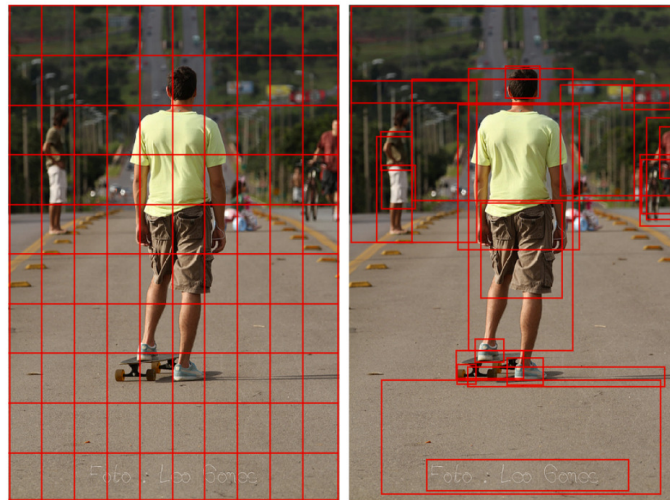


Figure 2.13: Attention models often operate on CNN features based on a uniform grid of equally-sized image regions (left). Another approach enables attention to be calculated at the salient image regions (right) [1].

## 2.4 Technology for the Blind

Computer vision can benefit many people. Especially those whose vision is very poor or non-existent. The approach to using technologies, such as computers or smartphones, for the blind and partially sighted (BVI) is significantly different from all others. For example, blind people usually do not use a computer mouse, because since they are not able to see the cursor, using a mouse would be counterproductive. This section describes the means of using technologies by visually impaired people.

### Braille Display

Some of the blind people are using an electronic refreshable device braille display (Figure 2.14) that allows a blind person to read or write the text as their main method for processing information, on the other hand, this device cost often more than \$500 and many people are not able to afford to buy it. Other people are not using because in opposite to 1960 when 50% of blind students were literate in Braille, based on 2016 statistics<sup>7</sup> 18.3% of students are learning the braille reading basics, and only 8.5% identify themselves as braille readers.

<sup>7</sup><https://brailleworks.com/braille-literacy-statistics/>

<sup>8</sup>Product website: <https://www.orbitresearch.com/product/orbit-reader-20/>



Figure 2.14: A refreshable braille display Orbit Reader 20<sup>8</sup>.

## Computers and Smartphones

The main technology that helps a blind person use a computer or smartphone is the screen reader. A screen reader is an application that helps the user with the orientation and processing of written text by screen reading. Users are often using 3rd party software JAWS<sup>9</sup> or NVDA<sup>10</sup> but can also use integrated Narrator for Windows users and VoiceOver for MacOS.

Widely used are computer keyboards for their low price and minor differences between multiple devices. A combination of screen reader and keyboard allows users to use a computer similarly to sighted people. The basic controls for websites consist of arrows, tab, and a few other navigation keys (to move from title to text, etc.). What was found is that blind people are trained to process audible information from screen reader very fast. For example, a sports commentator may be able to speak at a pace of 10 syllables per second, which is a limit for most people to comprehend. On the other hand, the trained blind person can process up to 25 syllables<sup>11</sup> per second, which allows him to consume content much faster than a sighted person.

In the case of smartphones<sup>12</sup>, BVI use accessibility settings such as TalkBack on Android or VoiceOver on iOS. These serve as screen readers but also change the preset gestures. The basics are reading the name of the item where the user places the finger, swipe left or right for the next or previous item, swipe up and down usually changes the type of items being scrolled, and three fingers are required to scroll.

Although blind people can sometimes be more effective than sighted people, they still face numerous daily challenges. One of these is the pop-ups on websites. If one window such as the remainder of cookies can not be closed easily, it can prevent a blind person from using the website at all.

---

<sup>9</sup><https://www.freedomscientific.com/products/software/jaws/>

<sup>10</sup><https://www.nvaccess.org/>

<sup>11</sup><https://www.scientificamerican.com/article/why-can-some-blind-people-process/>

<sup>12</sup>How do blind people use smartphones? <https://www.youtube.com/watch?v=IkQk8ZbToNo>

## 2.5 Fleiss' Kappa

The Fleiss' kappa [10] is a statistic measure of nominal or binary scale agreement between a fixed number of two or more raters. It expresses the extent to which the agreement between raters exceeds what would be expected if all raters evaluated completely randomly. If a fixed number of people assign a numerical rating to a set of items, then kappa can provide an extent of rating consistency.

The kappa denoted as  $\kappa$ , ranges from  $\kappa \leq 0$  if there is no agreement among the raters, to  $\kappa = 1$  for complete agreement. These values can be interpreted as in Table 2.2.  $\bar{P} - \bar{P}_e$  represents actually achieved agreement in excess of chance, and the factor  $1 - \bar{P}_e$  measures the extent of attainable agreement over and above what would be predicted by chance. Then the  $\kappa$  is defined as,

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}. \quad (2.12)$$

Agreement level	Poor	Slight	Fair	Moderate	Substantial	Almost perfect
$\kappa$	$\leq 0$	$\leq 0.2$	$\leq 0.4$	$\leq 0.6$	$\leq 0.8$	$\leq 1$

Table 2.2: Possible interpretation of  $\kappa$  values based on [35].

The total number of subjects is denoted as  $N$ , the number of categories  $k$ , and the number of ratings per subject  $n$ . The subscript  $i$ , where  $i = 1, \dots, N$  represents the subjects and the subscript  $j$ , where  $j = 1, \dots, k$  constitute the categories of the scale. Denoted as  $n_{ij}$  is a number of raters who assigned the  $i$ th item to the  $j$ th category, then proportion of all assignments to the  $j$ th category is the quantity  $p_j$ . Since  $\sum_j n_{ij} = n$ , thus  $\sum_j p_j = 1$ .

$$p_j = \frac{1}{Nn} \sum_{i=1}^N n_{ij}. \quad (2.13)$$

The degree of agreement between  $n$  raters for the  $i$ th item is indexed by the proportion  $P_i$  of agreeing on pairs out of the  $n(n-1)$  possible pairs. Then, the overall degree of agreement is denoted as  $\bar{P}$ . Agreement to some extent is solely expected based on chance.  $\bar{P}_e$  is the mean proportion of agreement if the raters acted purely at random.

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^k n_{ij}(n_{ij} - 1), \quad (2.14)$$

$$\bar{P} = \frac{1}{N} \sum_{i=1}^N P_i. \quad (2.15)$$

$$\bar{P}_e = \sum_{j=1}^k p_j^2. \quad (2.16)$$



## Chapter 3

# Visual Question Answering

The visual question answering (VQA) is the task of answering an open-ended<sup>1</sup> natural language question about a given image (Figure 3.1). The origin of this task is the VQA challenge 2016 based on the VQA dataset [2]. Similarly to image captioning, VQA expands into two computer science areas. The first one is Computer Vision (Section 2.3), where object detection is required to understand the context of a given image. On the other hand, an understanding of the language structure to be able to process the textual question and generate an answer representing the Natural Language Processing (Section 2.1). Visual questions target different images areas, including background details or underlying context. Therefore, a VQA system needs a more detailed understanding of an image and more complex reasoning than a system producing generic image captions.




Figure 3.1: Images and questions from authors of the VQA challenge (Source [2]).

<sup>1</sup>An open-ended question in the context of VQA is a question where the answer is a free-form text generated from the tokens from a vocabulary rather than choosing from a subset of possible answers.

The first Visual Question Answering challenge allowed an open-ended and multiple-choice approach (Figure 3.2). For the **open-ended** task, there are no possible answers given, therefore the system must construct the answer by itself. On the other hand, for the **multiple-choice** task there is a set of 18 predefined answers, and the goal is to pick the correct one. This task results in a higher answering performance. The possible answers for each question, with just one of them being correct, are set up as follows:

- The most common ground truth answer for the question
- Human created without seeing the image three plausible but wrong answers
- 10 of the most popular answers from the entire dataset
- The four others are taken randomly from all of the answers



**Open-Ended**

Q: What color scarf does the girl on the right have?

**Multiple-Choice**

Q: What color scarf does the girl on the right have?  
Multiple-Choice Options:

<ul style="list-style-type: none"> <li>(a) 625</li> <li>(b) gutter</li> <li>(c) red</li> <li>(d) green</li> <li>(e) white</li> <li>(f) 1</li> <li>(g) 3</li> <li>(h) hoodie</li> <li>(i) beige</li> </ul>	<ul style="list-style-type: none"> <li>(j) cowboy hats</li> <li>(k) pink</li> <li>(l) blue</li> <li>(m) 4</li> <li>(n) black</li> <li>(o) lucky tattoo</li> <li>(p) no</li> <li>(q) 2</li> <li>(r) yes</li> </ul>
---	---

**Ground-Truth**

Q: What color scarf does the girl on the right have?  
Ground-Truth Answers:

<ul style="list-style-type: none"> <li>(1) red</li> <li>(2) red and white</li> <li>(3) red</li> <li>(4) red</li> <li>(5) red</li> </ul>	<ul style="list-style-type: none"> <li>(6) red</li> <li>(7) green</li> <li>(8) red</li> <li>(9) red, white</li> <li>(10) red</li> </ul>
---	---

Figure 3.2: An example of image from VQA dataset (Source [2]).

## Interpretability and Bias Problem

The majority of machine learning systems including VQA suffer from interpretability. The reasoning in deep neural networks is distributed across millions of parameters, thus it is difficult for humans to understand the outputs of deep learning models. Understanding the process by which VQA models arrive at their decisions is an important mechanism of verifying that these models learn the knowledge that we would like them to learn. Interpretability is important for establishing whether a system is robust to biases that may exist in its training data. The bias issue is that VQA datasets tend to contain superficial regularities that allow models to memorize relationships between question and answer words. For example, if a model trained on [17] dataset receives a question asking “What sport is this?” it is very likely to answer “tennis” because it is the most represented sport in the dataset. Another bias, the phenomenon of visual priming can be observed from creating questions. In the [17] the correct answer for yes or no question is yes in 87% of cases. These biases can be exploited by researchers to achieve higher performance [18].

### 3.1 Datasets and Evaluation

Over the last years, there have been many efforts in comparing existing datasets [63, 30, 57]. This section will be covered those, with the greatest impact on the VQA task.

Microsoft Common Objects in Context (**MS-COCO**) [39] is not exactly a VQA dataset, but due to its large corpus of images, it is used by many VQA datasets. MS-COCO contains 91 common object categories (for instance: person, umbrella, tie, oven, train, horse, spoon, etc.) with 82 of them having more than 5,000 labelled instances. In total, there is a 2,500,000 labeled instances in 328,000 images (Figure 3.3). All those labels were added manually by humans and to each of the images were created five captions. The time required to annotate this dataset is estimated at more than 70,000 working hours for all people combined.



Figure 3.3: Authors of MS-COCO focused on finding primarily non-iconic images (c) (Source [39]).

The **DAQUAR** Dataset for Question Answering on Real-world images [41] is considered to be the first benchmark for VQA. It is based on images from NYU-Depth V2 [54] dataset and contains 1449 images (Figure 3.5). All pixels of images are labelled with one of the 894 classes. There is a total of 12468 questions, generated either automatically using 9 templates or annotated by humans. The authors propose two evaluation metrics. A simple accuracy and WUPS score, which calculates the similarity between two words based on their longest common subsequence in the taxonomy tree [63].

The **VQA** dataset [2] is a collection of real images, abstract scenes, questions and answers. There is a corpus of 204,721 real images from MS-COCO and 50K abstract scenes for exploring high-level reasoning. While real images are captions taken from MS-COCO, abstract scenes are generated artificially. For each image/scene were gathered three questions from unique workers, 0.76M in total. The questions and their answers were crowdsourced and can vary a lot. They range from knowledge base reasoning (“is this a vegetarian meal?”), concept detection (“how many fruits are in this picture”) to activity recognition (“is this man crying?”). Participants have presented an image and were asked to create a relevant question to the image. Afterwards, ten other participants were asked to answer the question. Based on statistics from authors, 38% of all questions are yes or no questions and 12% are number questions. The evaluation metric (Equation 3.1) considers multiple ground-truth answers. If the predicted answer was given by three or more (out



of 10) human annotators the answer is marked as correct. In case of less than three is an accuracy calculated accordingly.

$$\text{accuracy} = \min\left(\frac{\# \text{ GT answers same as predicted answer}}{3}, 1\right) \quad (3.1)$$

The **Visual Madlibs** dataset [69] is also based on images from MS-COCO. There is 360,001 focused descriptions for 10,738 images. The evaluation tasks are either fill-in-the-blank (a strategy for collecting captions) and a multiple-choice question answering (Figure 3.4). For the first case annotators were presented with an image and a fill-in-the-blank template for instance “The banana is [blank]” and asked to fill in the [blank] with a description of the appearance (or any other attribute) of banana. There were 12 types of the question considering image’s scene, emotion, interesting, past, future, object’s attribute, affordance, position, person’s attribute, activity, location, and relation of pairs. For the other task, the computer is provided with an image and a partial description such as “The person is [blank]”. Four plausible choices are provided out of which is only one correct.

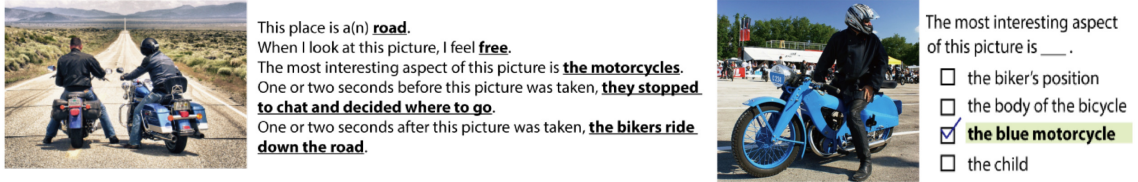


Figure 3.4: An example of Madlibs fill-in-the-blank (left) and multiple-choice (right) (Source [69]).

The **COCO-QA** [50] uses images and captions from MS-COCO dataset [39] to artificially generate questions based on the language model LSTM. The questions are of four types. The Object questions are asking about objects using “what”. Others are the number, color and location questions (Figure 3.5). The metrics are identical to DAQUAR. The dataset contains a total of 123,287 images, one question to each of the images, and answers are all single word.



Figure 3.5: An example of COCO-QA questions (Source [50]).

The **Visul Genome** (VG) [33] contains 108,077 images from MS-COCO and 1,700,000 question-answer pairs with average of 16.40 QA pairs per image (Figure 3.6). Unlike previous datasets, which were collected for a single task, the Visual Genome dataset was collected

to be a general-purpose representation of the visual world, without bias toward a particular task. The VG is based on six types of questions: what, where, how, when, who and why. It contains approximately 35 objects, 26 attributes and 21 relationships per image exceeding others by a large margin, thus has better answer diversity in comparison to other datasets. There are no binary questions, and 57% of the answers are single words. The model is correct on a QA if one of the predictions matches exactly with the ground-truth answer for that question. This evaluation method works well when the answers are short. Human performance was also reported on these questions by presenting them with the image question pair along with 10 multiple choice answers out of which one was the ground truth and the other 9 were randomly chosen from the dataset [30].

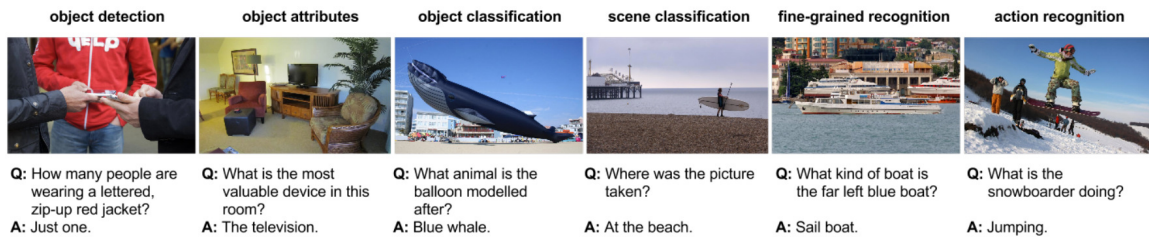


Figure 3.6: An example of the Visual Genome QA pairs (Source [33]).

The **Visual7W** [73] is based on the VG and contains 47,300 images. The name stands for 7 'w' words, expanding VG with 'which'. The 'what', 'who' and 'how' questions often relate to recognition tasks with spatial reasoning. The 'where', 'when' and 'why' on the other hand, usually involve high-level common sense reasoning. This improvement is used for the selection of correct bounding box, therefore, linking object mentions to their bounding box in the image (Figure 3.7). The evaluation for multiple-choice is the same as for [69]. The objects mentioned in the QA pairs are grounded to their corresponding bounding boxes in the images. The groundings enable examining the object distributions and resolve the coreference ambiguity [63].

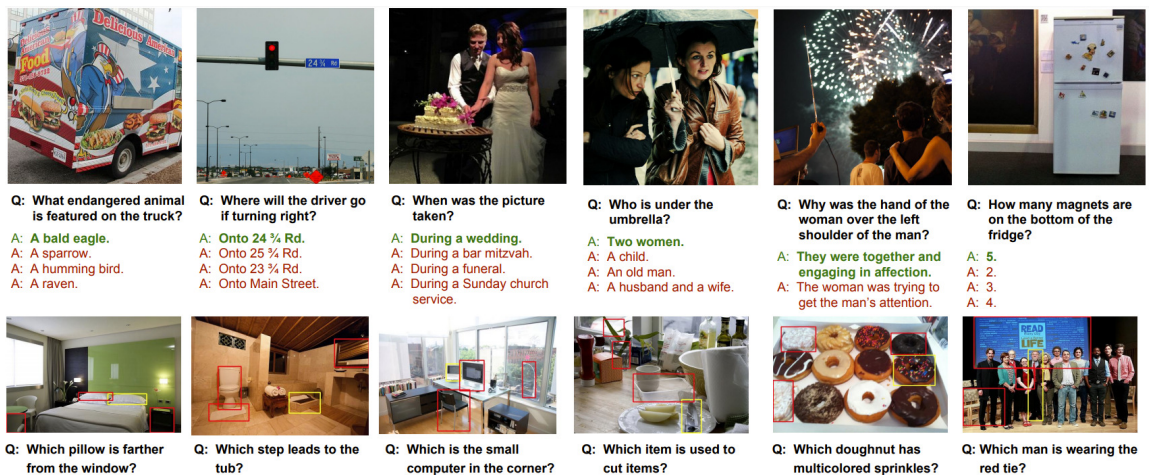


Figure 3.7: Examples of multiple-choice QA. The first row shows telling questions with one correct and others wrong answers. In the second row are pointing (which) questions where the correct answer is the yellow box and the red boxes are wrong answers (Source [73]).

The **VQAv2** [17] addresses the problem of ignoring most of the visual information. For each tuple of image question and answer another image was found, where the answer for the same question would be different (see Figure 3.8). There is approximately twice the number of image-question pairs and is added about 330K binary abstract scenes in comparison with VQAv1.

The dataset is overall more balanced with significantly reduced language biases in comparison with its predecessor. Authors noticed that when the models were trained on an unbalanced VQA dataset and tested also on an unbalanced VQA dataset, the model’s performance was more than by 10% better on yes/no question type when compared to training on unbalanced VQA, but testing on balanced VQAv2 dataset. This suggests that the models are really exploiting the language biases, which leads to high accuracy on an unbalanced dataset because that dataset also contains these biases. The evaluation metric is same as for the VQAv1 (Equation 3.1). The accuracy on the test-standard split of this dataset is the primary metric used by recent methods for evaluation.



Figure 3.8: An example of question-images pairs from VQAv2 dataset (Source [17]).

The **GQA** [25] consists of generated 22,669,678 questions and 113,018 images. Each image is annotated with a Scene Graph representing the objects, attributes, and relations. Each question has a functional program, which lists reasoning steps to arrive at the answer. Construction process is visualised in Figure 3.9. The dataset has a vocabulary size of 3097 words and 1878 possible answers. The GQA proposes five new metrics.

**Consistency** measures responses consistency across different questions. The model should not contradict its own answer when being presented with a question regarding its previous answer. For example, if the apple is identified as “red” in previous answer about the same object, the answer next time should not be “green”. For each QA pair  $(q, a)$  is defined as a set of entailed questions, the answers to which can be unambiguously derived from  $(q, a)$ . The accuracy is measured for each question the model answered correctly with its entailed questions. These scores are then averaged across all correctly answered questions.

**Validity** examines whether a given answer is in the question scope, for instance answering some colour to a colour question.

**Plausibility** measures whether the answer makes sense by checking the whole dataset if the question’s subject occurs at least once in relation to the answer.

**Distribution** measures match between true answer and model predicted distribution using the Chi-Square [11] statistic. It shows whether the model also predicts the less frequent answers.

**Grounding** checks whether the model attends relevant regions of the image to a given question. A pointer  $r$  to the visual region to which the question or answer refers is defined for each dataset instance. For this region is measured the model’s probability.

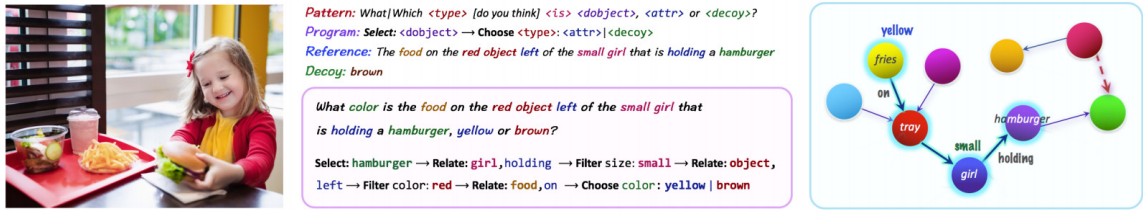


Figure 3.9: In GQA dataset each question is represented in natural language and a functional program (Source [25]).

### The Summary of VQA datasets

A recent trend is using a combination of multiple datasets for pre-training for a purpose of creating a corpus as big as possible. Many datasets were published, but most of them are rarely used. The widely used are VQAv2 and GQA. The GQA was used three times more often in the year 2019 than VQAv2, while in the year 2020 these two datasets were used just about the same<sup>2</sup>. Although GQA contains just 1878 possible answers, it covers 88.8% questions and 70.6% answers of the VQAv2 dataset. It is clear from Table 3.1, that recent datasets use more images, and also way more questions than before.

	Number of Images	Number of Questions	Avg. questions per Images	Avg. question Length
DAQUAR	1,449	12,468	8.60	11.5
COCO-QA	117,684	117,684	1.00	9.7
Visual Madlibs	10,738	360,001	33.52	4.9
Visual7W	47,300	327,939	6.93	6.9
VQAv2	286,046	1,289,287	5.40	8.1
GQA	113,018	22,669,678	200.58	11.0

Table 3.1: Comparison of datasets for VQA inspired by [63].

<sup>2</sup>These statistics are based on the data from <https://paperswithcode.com/>.



## 3.2 Methods

The methods in this section achieved state-of-the-art (SoTA) performance and most of them are publicly available. Most of the time, it does not take more than a few months for new model to outperform its predecessor.

**Pythia** [29] is the winning entry of the VQA challenge 2018 and forms the basis of modular multimodal framework [56] (MMF). Implementation is based on the up-down object detection model [1]. Image features are detected by using Faster R-CNN [51] pre-trained on the Visual Genome dataset [33] and the ResNet-152 [65] network was chosen as the backbone network. Each region is represented by a 2048-dimensional feature after average pooling from a  $7 \times 7$  grid. question text is used for computing the top-down attention for each object in the image. Number of object proposals is fixed at 100 for all images. The model implements multimodal embedding of the question and the image followed by a prediction of regression of scores over a set of candidate answers. Each of tanh layers implements a function  $f_a : x \in \mathbb{R}^m \rightarrow y \in \mathbb{R}^n$  with parameters  $W, W', b, b'$ .

$$\tilde{y} = \tanh(Wx + b) \quad (3.2)$$

$$g = \sigma(W'x + b') \quad (3.3)$$

$$y = \tilde{y} \circ g \quad (3.4)$$

$W, W' \in \mathbb{R}^{n \times m}$  are learned weights,  $b, b' \in \mathbb{R}^n$  are learned biases and  $\circ$  is Hadamard product. The vector  $g$  acts multiplicatively as a gate on the activation  $\tilde{y}$  and  $\sigma$  is the sigmoid activation function. Each question is encoded as the last hidden state  $q$  of a GRU [3], with each input word represented by the learned word embedding. For each location  $i = 1 \dots k$  in the image, the feature vector  $v_i$  is concatenated with question embedding  $q$  to generate attention weights  $a_i$ , where  $w_a^T$  is a learned parameter vector. The attention weights are normalized over all locations with a softmax function. The 2048-dimensional vector  $\hat{v}$  represents the attended image.

$$a_i = w_a^T f_a([v_i, q]) \quad (3.5)$$

$$\alpha = \text{softmax}(a) \quad (3.6)$$

$$\hat{v} = \sum_{i=1}^K \alpha_i v_i \quad (3.7)$$

Vector  $h$  is calculated from representations of the question ( $q$ ) and the image ( $\hat{v}$ ) passed through the non-linear layers ( $f_q$  and  $f_v$ ) combined by using the Hadamard product. The probability distribution  $p(y)$  over possible outputs  $y$  is calculated with learned weights  $W_o \in \mathbb{R}^{|\Sigma| \times M}$  of vocabulary  $\Sigma$ .

$$h = f_q(q) \circ f_v(\hat{v}) \quad (3.8)$$

$$p(y) = \sigma(W_o f_o(h)) \quad (3.9)$$

For fine-tuning is used object detector based on feature pyramid networks from Detectron [14], which is based on ResNeXt [65] backbone with two fully connected layers (fc6 and fc7) for region classification. That allows to extract 2048-dimensional fc6 features and fine-tune the fc7 parameters, which requires significantly less computation. The model achieved final performance on test-standard VQAv2 72.27% [29].

**Object-Semantics Aligned Pre-training for Vision-Language Tasks [38] (Oscar)** is based on observation, that objects mentioned in the text can be accurately detected, therefore, authors present tags as anchor points (Figure 3.10). For instance on the MS-COCO [39] the percentages that an image and corresponding text share at least 1,2,3 objects are 49.7%, 22.2% and 12.9%. Model is pre-trained on the corpus of 6.5 million text-image pairs and then fine-tuned for a specific task, such as image captioning or visual question answering.

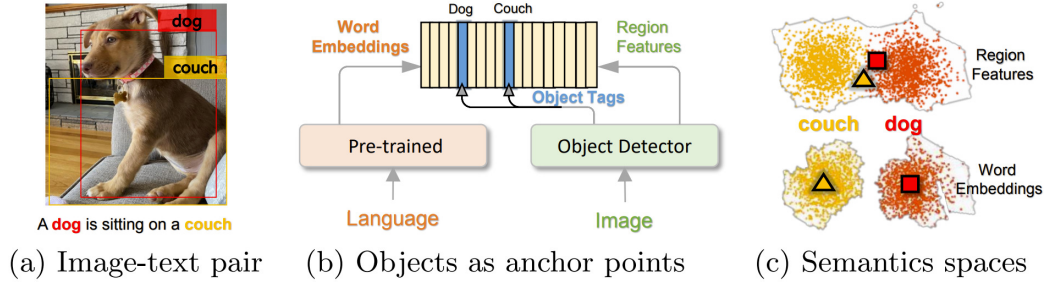


Figure 3.10: (a) An example of input image and a caption (b) visualization of object tags (c) even though dog and couch regions overlap in the visual feature space, the word embeddings are distinctive (Source [38]).

The triples  $(w, q, v)$  are used as input. These are composed of a sequence of text embedding  $w$ , object tags  $q$  detected from image and image region vectors  $v$ . While existing methods represent input as  $(w, v)$ , to ease the learning of image-text alignments Oscar introduces  $q$  as anchor points. The image regions from which are the  $q$  detected are likely to have higher attention weights than other regions. Pre-trained BERT [6] is used to identify alignments between  $q$  and  $w$ . Model architecture is visualised in Figure 3.11. Visual semantics  $(v', z)$  and a sequence of object tags  $q$  are extracted with Faster R-CNN [51]. Region feature  $v' \in \mathbb{R}^P$  is a vector of  $P$  dimensions ( $P = 2048$ ) and  $z$  are coordinates of  $R$  dimensions ( $R = 4$ ).  $v'$  and  $z$  are concatenated and transformed into  $v$  by using a linear projection.

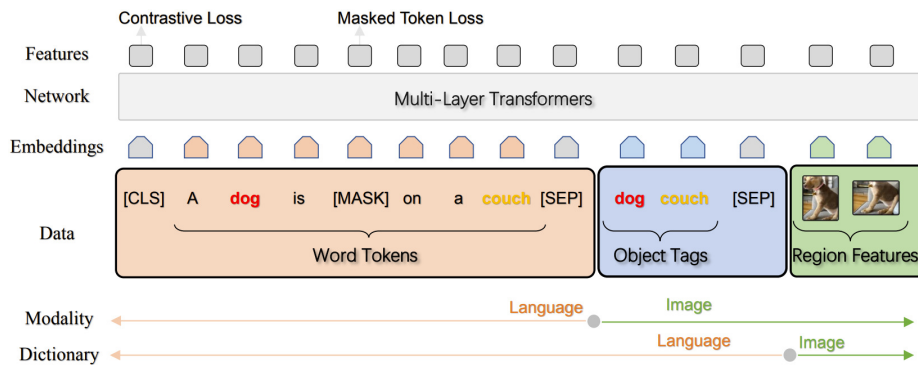


Figure 3.11: Input of triples, word-tag-region is fed into the BERT. Dictionary view differentiates two semantic spaces and is represented by masked token loss. Modality view distinguishes between text and an image and is represented by contrastive loss (Source [38]).

There are two perspectives for pre-training, a dictionary view with masked token loss (MTL) and a modality view with contrastive loss (CL). The pre-training objective is a sum

of these losses (Equation 3.10). Authors claim that based on their experiments Oscar yields superior performance in comparison with other existing methods even though the overall loss is much simpler.

$$\mathcal{L} = \mathcal{L}_{\text{MTL}} + \mathcal{L}_{\text{C}} \quad (3.10)$$

For the dictionary view, the object tags and word tokens share the same linguistic semantic space. MTL, similar to the masked language model used by BERT is applied to the discrete token sequence defined as  $h \triangleq [w, q]$ , where “,” stands for concatenation. At each iteration input tokens,  $h_i$  are masked with a probability of 15% and the goal is to predict masked word based on adjacent tokens  $h_{\setminus i}$  and image features  $v$ , minimizing the negative log-likelihood.

$$\mathcal{L}_{\text{MTL}} = -\mathbb{E}_{(v,h) \sim D} \log p(h_i | h_{\setminus i}, v) \quad (3.11)$$

In case of modality view each input triple is split into image modality  $h' \triangleq [q, v]$  and  $w$  for language modality. The contrastive loss (Equation 3.12) replaces tag with randomly chosen different tag from the dataset with a probability of 50%. Fully connected layer with binary classifier  $f(\cdot)$  predict if the tag is original ( $y = 1$ ) or not ( $y = 0$ ).

$$\mathcal{L}_{\text{C}} = -\mathbb{E}_{(h',w) \sim D} \log p(y | f(h', w)) \quad (3.12)$$

The architecture is based on two variants of BERT [6] with different hidden sizes 768 and 1024 for the base and large model, respectively. During inference for image captioning are encoded as input image regions, object tags, and a special token CLS. The process of generating starts by feeding in a MASK token and sampling a token from the vocabulary based on the likelihood output. The MASK token used in the previous input sequence is replaced with the sampled token and then a new [MASK] is appended for the next word prediction. The output of STOP token terminates the generating For VQA the model is trained on the VQAv2 dataset and the task is treated as multi-label classification<sup>3</sup> problems. The achieved score for the large model is 73.82% on test-std of the VQAv2 dataset.

**Bilinear Graph Networks** (BGN) for Visual Question Answering [44] are based on cooperating layers of image-graph and question-graph which leads to the realization of multi-step reasoning. The goal of the **image-graph** is to locate the objects related to semantic information of each word in the question. The node of the graph is defined  $\mathcal{V} = Q \cup V$ , where  $V$  are the visual features of the detected objects and  $Q$  are textual features of the question. The edges of the graph are the computed graph weights based on  $Q$  and  $V$ . The **question-graph** amplifies the implicit relationships between objects by exploiting information across different embeddings. The question-graph nodes are the output nodes of the image-graph and the graph weights are the self-attention of inputs. The combination of these two graphs allows solving complex and compositional questions. Model architecture is visualised in Figure 3.12. The ablation studies showed that BGN significantly outperforms other graph-based methods. The BGN model achieved an accuracy of 72.41% on the test-std VQAv2 dataset.

---

<sup>3</sup>Multi-label classification is a classification problem where multiple labels may be assigned to each instance, however, there is no constraint on how many of the classes the instance can be assigned to.

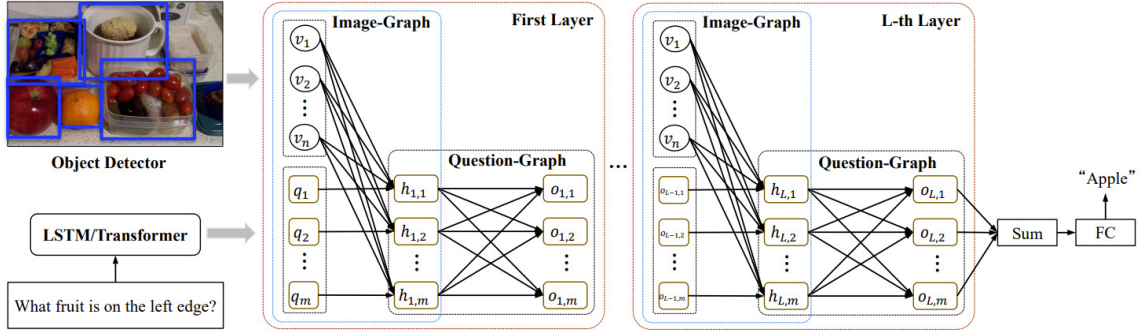


Figure 3.12: The architecture of the Bilinear Graph Networks. Basic module of BGN is composed of one image-graph following one question-graph and these modules are then stacked to create multiple layers (Source [44]).

In **Grid Features** [28] authors found that grid-based convolutional features can match the accuracy of the most used region-based [1] features in a fraction of computational time. For example, when ResNet [20] was used as a backbone, the computation was more than forty times faster. Their ablation analysis suggests that a large-scale annotated dataset for pre-training and high spatial resolution of the input images is much more important than the type of features. Grid Features use 1x1 RoIPool instead of 14x14 RoIPool from the Faster R-CNN [51], thus representing each region with a single vector, rather than a three-dimensional tensor. This means each vector on the grid feature map is forced to cover all the information for a spatial region, which can result in stronger grid features. Because a pre-trained convolutional network works best with inputs of particular spatial dimensions, two fully connected layers are added on the top to accept vectors as input. Authors of the Grid Features achieved an accuracy of 72.71 on the test-std VQAv2 dataset.

**Oscar+** [72] model is improved continuation of Oscar. The pre-training corpus is based on three types of datasets. Human-annotated image captioning datasets with generated image tags such as MS-COCO, VQA datasets such as VQAv2 with questions and human-annotated answers and image tagging datasets with generated captions and human-annotated tags. New object detection model, that produces better visual features of images than previous was developed. The large-scale object-attribute detection model is based on the ResNeXt-152 C4 architecture. For pre-training objectives, Masked Token Loss is the same as for Oscar and the other one is the 3-way Contrastive Loss, which considers both, (caption, tags, image-features) for image captioning and (question, answer, image-features) for VQA data. Negative examples are constructed for polluted captions and polluted answers. The polluted captions and answers are uniformly sampled from all (captions and answers) in the corpus. Therefore, the dataset contains 50% correct triples, 25% polluted captions and 25% polluted answers. The results of oscar+ ablation analysis are that the improvement of VQA is a compound of increasing model size and dataset size. At the time of publishing, authors achieved a performance of 76.60% on test-std of VQAv2 dataset.



# Chapter 4

## Usage Scenarios

Although Visual Question Answering (VQA) has been studied by many researchers the actual usage outside research might be unclear. Where could VQA be useful? There are many possible applications but just a few of those could be truly useful. To name a few, it could be useful in retrieval systems such as searching in maps. Even though, this work did not find any such application in practice. It could be used for example for answering questions about specific objects. Similar to this use case, VQA could be useful in data analysis or searching for specific information in huge amounts of images such as video surveillance. There are other possibilities, where it could actually help. Medical personnel could benefit from another point of view at X-rays and other images. Other people who could benefit from this technology are blind and visually impaired (BVI). This chapter presents a case study to determine if and how the VQA could be used inside the BVI community. The first Section 4.1 examines smartphone applications used by BVI. Then is described a demonstrative application (Section 4.2). The following Section 4.3 describes the process of finding BVI to use an application and then is discussed knowledge gathered by a questionnaire. The last Section 4.4 of this chapter deals with the creation of a smartphone application.

### 4.1 Existing Applications

There are many smartphone applications for BVI trying to make their lives easier. The focus of these applications could be divided into one of the following categories. The most significant category is connecting BVI to sighted operators or volunteers for assistance such as *Blind*<sup>1</sup> or *Be My Eyes*<sup>2</sup> with millions of downloads available for both Android and iOS. Although they are the most used, they are not based on machine learning, thus they are not so important for this work. Another category could be navigation around the city. The most often used are built-in Google and Apple maps based on the smartphone operating system. No application providing VQA was found. The last category consists of image captioning, which is similar to VQA and other similar functionalities based on machine learning. Some of these functionalities are reading text, face recognition or recognition of paper money value. Based on personal experience, these applications are not able to recognize other money than US dollars. Most image captioning applications have less than a few thousand downloads. To gain a qualitative assessment of such applications, this work

---

<sup>1</sup>[https://play.google.com/store/apps/details?id=com.teamblind.blind&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.teamblind.blind&hl=en_US&gl=US)

<sup>2</sup>[https://play.google.com/store/apps/details?id=com.bemyeyes.bemyeyes&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.bemyeyes.bemyeyes&hl=en_US&gl=US)

covers those with more than 50,000 downloads.

The *Google Lookout*<sup>3</sup> is available on Android for free. It allows real-time object recognition, reading documents and recognition of paper money. In addition, this application allows retrieving data about food based on its barcode.

The *Seeing AI*<sup>4</sup> is available on iOS also for free. In comparison with Lookout, this application allows to add and recognize faces. The rest of the applications are both on Android and iOS.

The *TapTapSee*<sup>5</sup> is available for free and its only feature is image captioning.

The *Sullivan+*<sup>6</sup> is free with occasional advertisements. There are similar functionalities to previous applications with the addition of colour recognition and a magnifier tool.

The *Supersense*<sup>7</sup> is free for reading simple text and finding specific items but allows a premium plan for 6 USD per month or 145 USD for a lifetime. This plan adds support for Hindu, Arabic and Russian languages, handwriting recognition or product barcode scanner.

The *Envision AI*<sup>8</sup> offers a free 14-day trial and then requires a subscription of 2 USD per month or 140 USD for a lifetime. The features (Figure 4.1) are very similar to Seeing AI with the added possibility to *ask for call*. It can take hours for the operator to call.

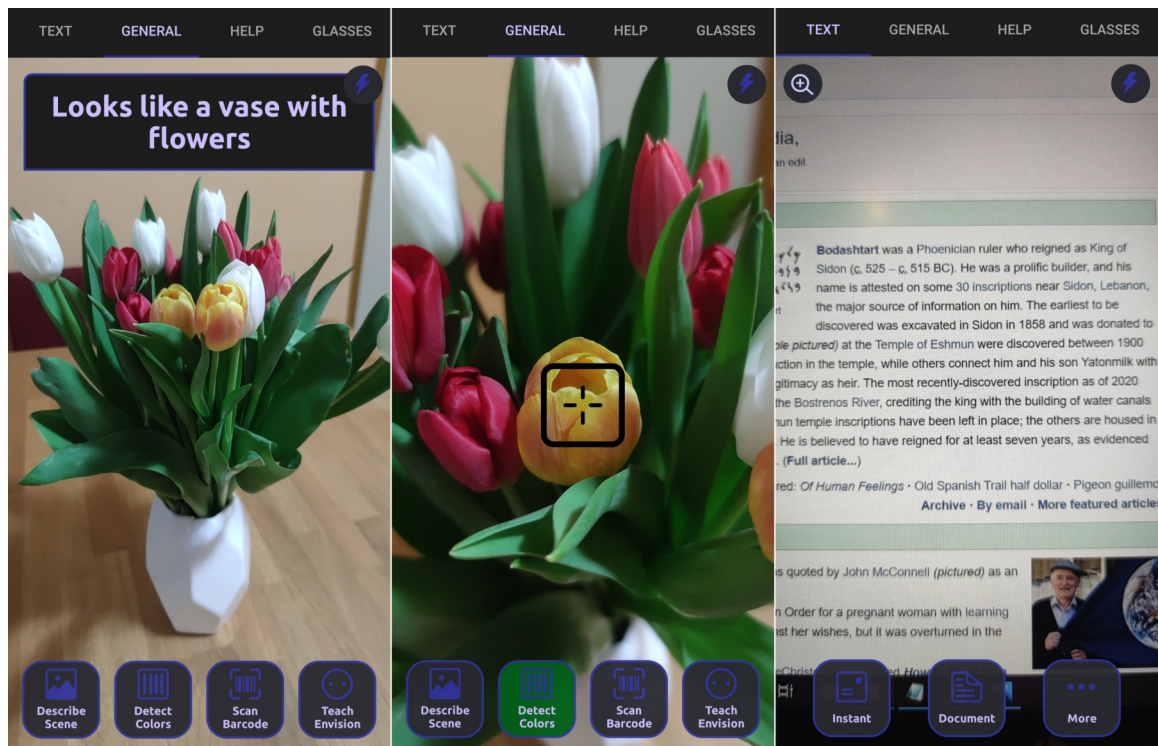


Figure 4.1: Image captioning, color detection and reading of text with Envision AI.

<sup>3</sup><https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.reveal&hl=cs&gl=US>

<sup>4</sup><https://www.microsoft.com/en-us/ai/seeing-ai>

<sup>5</sup><https://apps.apple.com/us/app/taptapsee-blind-visually-impaired/id567635020>

<sup>6</sup><https://play.google.com/store/apps/details?id=tuat.kr.sullivan&hl=cs&gl=US>

<sup>7</sup><https://play.google.com/store/apps/details?id=com.mediate.supersense&hl=cs&gl=US>

<sup>8</sup><https://www.letsenvision.com/envision-app>

## 4.2 Demonstrative Application

Such an application that would offer VQA or a combination of image captioning and VQA was not found. For this reason, I decided to create a demonstrative application<sup>9</sup> to find out if VQA can be beneficial for the BVI community. I choose to implement this application as a simple website for the following reasons. Installation of applications that are not listed in app stores requires unnecessary complexity. This could be solved by adding my application to these stores, but it would still require an extensive process, since I would like to be able to offer my app to users with iOS, HarmonyOS and any Android device. Another reason is that a website can be used not only with a smartphone but also with a computer.

The front end was implemented in an open-source JavaScript library React<sup>10</sup> since it is fast, flexible and easy to use. Uppy<sup>11</sup> was used for uploading images. It allows to choose an image from a gallery or take a photo on a smartphone. For communication of client-side and a VQA model was chosen a lightweight Python framework Flask. The VQA model I choose for this use case was Pythia described in Section 3.2 trained on the VQAv2 dataset, which contains a large number of images.

### User's View

The application (Figure 4.2) was created with emphasis for simplicity to be used easily with accessibility settings. The user visits a website and is presented with just one button. When pressed, the user is asked to either choose an image from the gallery or take a photo. The image is displayed<sup>12</sup>, a text box appears and focus<sup>13</sup> is set to this box for the user to be able to start typing his question. This question can be submitted by just pressing *enter* or by pressing the following *submit* button. When a user receives answers, the focus is set accordingly and the user is presented with the three most probable answers and their probabilities.

### Internal View

1. Client sends request to upload an image - (HTTP request: POST /upload-image).
2. Server saves the file in a temporary location.
3. Server calculates SHA256 hash from that file and renames it as such.
4. Server moves renamed file to a specified directory.
5. Server sends back a response with a **filename**.
6. Client receives a successful response, and a question field is made available.
7. After the user types in the question and presses “enter” new request is sent to the server - (HTTP request: POST /ask-question/**filename**), where **filename** is the response from upload request and in request's body is a question.

---

<sup>9</sup><http://vqa.wz.cz/>

<sup>10</sup><https://reactjs.org/>


<sup>11</sup><https://uppy.io/>

<sup>12</sup>Displaying image is useless for a blind person, but can be useful for colourblind or other impairments.

<sup>13</sup>By setting focus, the user's screen reader informs him where he is focused.


8. Server processes the question for stored image and responds with an array of answers and their probabilities.
9. Client shows the response to the user.
10. When a new question is entered for the same image steps 7-9 are repeated.
11. When a new image is uploaded process resets from step 1.

Upload your file:



Drop files here or [browse](#)

Uploaded image:



**Answers:**

- no (98%)
- yes (2%)
- i don't know (0%)

**Ask a question:**

**Submit**

Figure 4.2: An example of using demonstrative application.

### 4.3 Testing by Blind and Visually Impaired

The next step was to find blind and visually impaired people that could try my demonstrative application and provide valuable feedback. First of all I tried contacting the biggest organizations (European Blind Union<sup>14</sup>, American Council of the Blind<sup>15</sup>, etc.) that help BVI people all over the world and asking them if they could forward my message to anyone that could participate in my testing. With this approach, I was not able to obtain the required participants. I proceeded with my proposal towards few members of the Czech organization Sons<sup>16</sup>. This way I was able to get in touch with about a dozen of BVI and some were even willing to pass forward my demonstrative app to a friend or two. The instructions were to use the app for their struggles for up to two weeks and then fill out a simple questionnaire.

#### Feedback Obtained by Questionnaire

The motivation for this questionnaire was to find out if BVI people are interested in using VQA in their everyday life. What is the use case where it makes sense to them or if they are not interested, for what reason? While creating this questionnaire I tried to make it as simple and short as possible. The time required to fill out the form should not exceed five minutes. Most of the questions were of type *choose one* with an average of 4 possible answers. The questions of type *choose one or more* with the possibility to add own answer and the last question was the only one open-ended. This questionnaire was straightforward, so even non-technical BVI people were able to fill it out.

#### Population

The form was filled by a total of 20 people. On average, each participant used an application for 11 images with an average of 2.3 questions per image. Based on personal questions 70% of respondents are men and 60% are users of Apple smartphone. Figure 4.3 shows surprising results with the largest group of 40 to 49 years. The expected age of respondents was around 20 since younger people are generally more open to new technologies. An explanation for these results could be that loss of sight often comes with age.

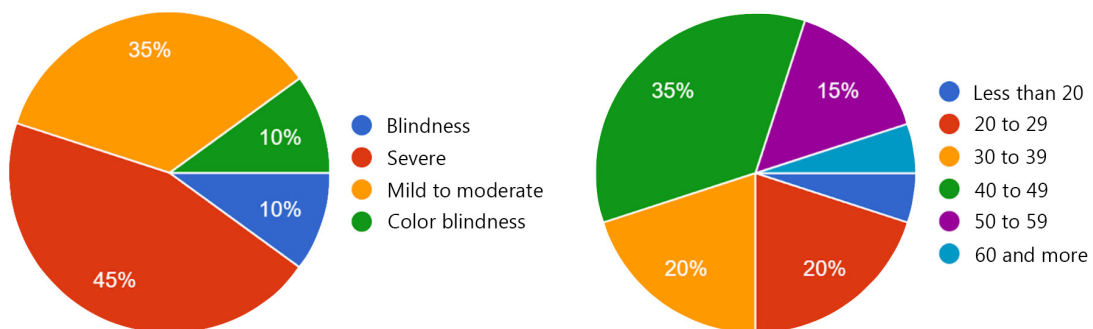


Figure 4.3: Proportions of participants with *severity* of visual impairment (left), *age* (right).

<sup>14</sup><http://www.euroblind.org/>

<sup>15</sup><https://www.acb.org/>

<sup>16</sup><https://www.sons.cz/>

More than half of the respondents are using image captioning software on their smartphones a few times a week. As can be seen in Figure 4.4 on the left more than half of the respondents to find VQA to be useful for them. on the other hand, none of the participants chose the *not useful* option. The graph on the right shows that most of the participants would appreciate it if their image captioning application on the smartphone also supported visual question answering.

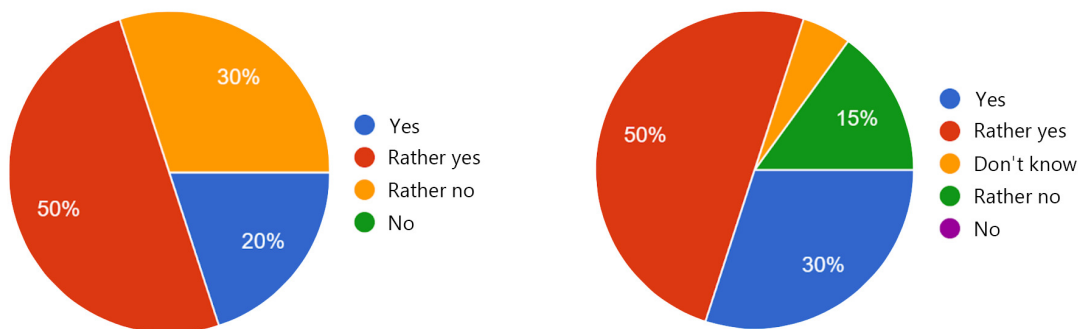


Figure 4.4: (left) Do you find visual question answering useful for you? (right) Would you appreciate if your image captioning smartphone application allowed also using visual question answering?

The goal of the following question (Figure 4.5) is to find out the use cases where does VQA make sense to participants. The results show that only 10% of respondents are not interested in any of the presented options, therefore, it can be assumed that the BVI community would appreciate a smartphone application with VQA. Based on this question alone this application could be aimed at either orientation, localization or could be specialized in the selection of clothes.

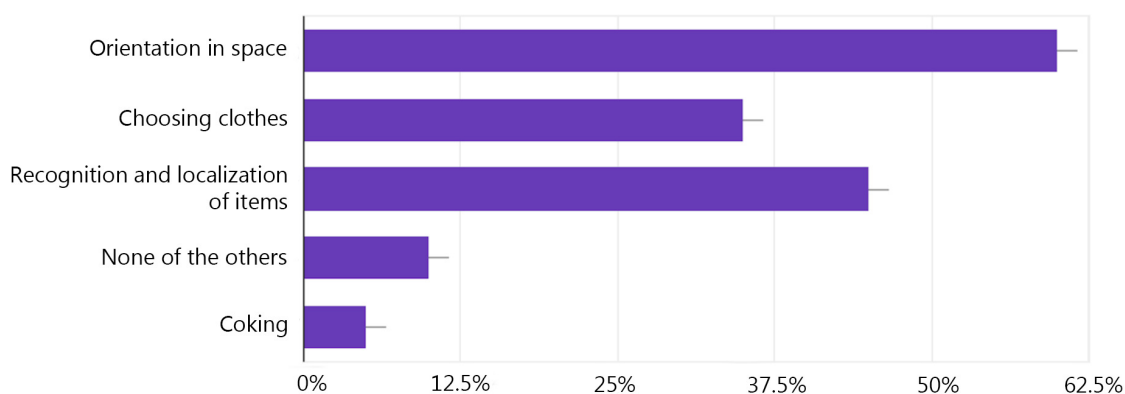


Figure 4.5: Where does using visual question answering make sense to you?



The most common answers for the next question (Figure 4.6) were either online shopping or none of the options. These results were discussed with a person from the community. Online shopping for a blind person is a challenging task. Choosing this option is motivated by descriptions of products online. These descriptions are often focused on technical details, but rarely describe the appearance of the product. For instance, sellers of furniture or clothes often rely on images. The description “black t-shirt with a graphic print” combined with the image is good enough for a sighted person, but for a blind person, it is not suitable. The VQA models described in this thesis could be used for such application, but it would require training on a specialized dataset.

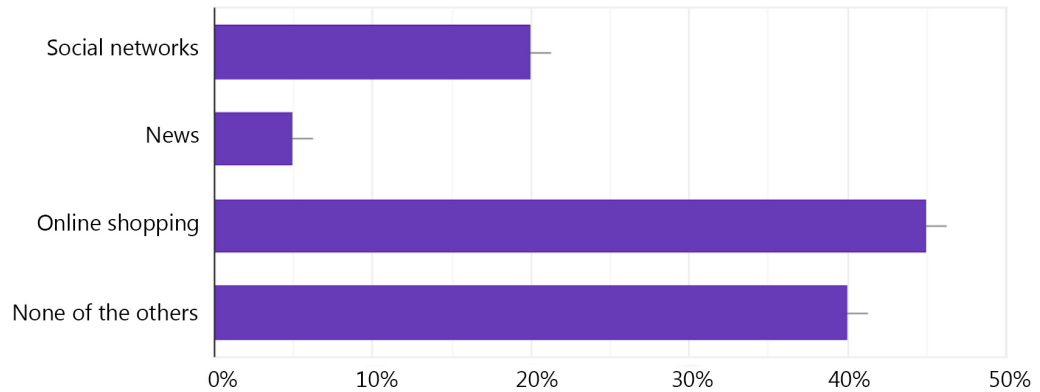


Figure 4.6: Where would you use VQA while working on a computer?

The following graph (Figure 4.7) shows the model’s accuracy and overall satisfaction with the application. 80% of participants rated the accuracy as *Sufficient* or better and the overall rating from 55% of participators was positive. The most common problem recognized by 35% of participants was the need for an internet connection. 30% of respondents did not fill any of the problems. Other difficulties were with the use of the application inside of the internet browser (25%). The other problems reported (5%) were an absence of the Czech language, results being too generic and the application being too slow. Another problem (5%) was a poor performance for questions regarding text in images. One of the suggestions was to focus on a specific problem such as reading digital displays or recognition of colour patterns on clothes.

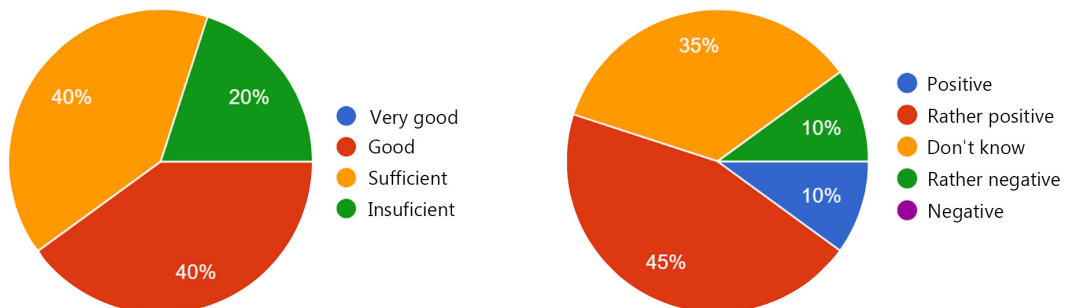


Figure 4.7: (left) Please rate the accuracy of results. (right) Overall rating of application.



## 4.4 Smartphone Application

Based on the previous section people of the BVI community would appreciate smartphone application with visual question answering. For this reason, an application on Android was created, which is described in this section. Since the target group for this application are primarily blind people, the visual aspect of this application is not of high importance. More emphasis is on simplicity and compatibility with the screen reader. The visual question answering model is exactly the same as was described in the demonstrative application (Section 4.2) and the overall design is also very similar since the application overall was rated by users positively.

### Implementation

For the creation of a mobile application, was not use any of the multiplatform frameworks and instead was chosen the native technology for the Android operating system. The application is supported by all smartphones with Android version 4.3 (API level 19) and higher. The programming language used in this case is Java.

The native material design<sup>17</sup> of version 1.3 without any modifications was used for visual representation since the visual aspect is not a priority for blind people. The second dependency is the Volley<sup>18</sup> library of version 1.2.0. Volley is a library for establishing and managing HTTP communication. The main advantage over Java HTTP client is the automatic scheduling of network requests and the support of multiple concurrent network connections.

Four different permissions are required for the application to run properly. These permissions are obtained during the use of the application. If the application requires one of these permissions, the user is offered a dialogue in which he can either accept this request or reject it. In case of rejection, it is not possible to continue. These permissions are required for opening the camera, storing, and reading from storage, and for sending requests to the VQA server. The required permissions are displayed below.

- `android.permission.CAMERA`
- `android.permission.WRITE_EXTERNAL_STORAGE`
- `android.permission.READ_EXTERNAL_STORAGE`
- `android.permission.INTERNET`

The whole application is based on only one android activity `MainActivity`. In this activity, there are two `Button` elements for capturing another image and sending a question. `EditText` element is used as input the question and `TextView` shows answers provided by VQA model. The activity also displays one dynamic `ImageView` element, which is used to display the last photo taken. The application was tested with Android built-in screen reader Talkback.

---

<sup>17</sup><https://material.io/develop/android>

<sup>18</sup><https://developer.android.com/training/volley>

## Workflow

After launching the application (Figure 4.8), the user is either prompted to authorize the permissions via dialogue or (if the authorization was granted in the past) the camera is displayed. Then, the user can take a photo in the same way as in the regular camera app, the controls are not altered. When the user is satisfied with the photo, it is possible to confirm the selection. The user is then redirected to the main screen of the application and in the background, the photo is sent to the server. The user is returned a hash file in the same way, as in the case of the web application (Section 4.2). The last photo taken is displayed and the user is presented with two options. If the user is a sighted person and is not satisfied with the photo, it is possible to click the “Capture image” button and repeat the whole process. Otherwise, the keyboard opens, and the user is offered a text field in which he can enter a question and confirm sending by pressing the “Ask question” button. The button is pressable only if the image was already successfully uploaded to the server, which is most of the time faster than typing the question. The confirmation sends another query to the server, which contains a hash of the last taken photo and a question (in the control element `EditText`). After obtaining a response, three of the most probable answers chosen by the VQA model are shown to the user in the form of a list, where each answer has a percentage representing the model’s confidence for that answer. At this point, as in the previous step, the user can select a new photo or enter a new question. The whole process of using the application is visualised in Figure 4.9.

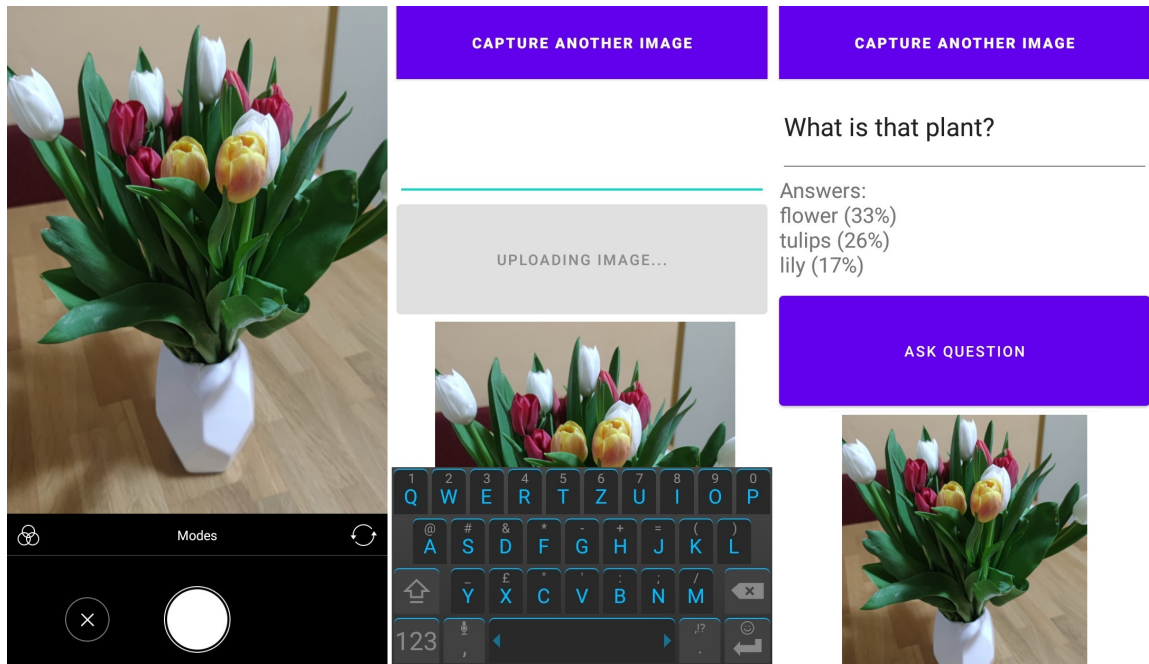


Figure 4.8: An example of using android application. Photo is taken, user enters his question, and the answers are displayed.

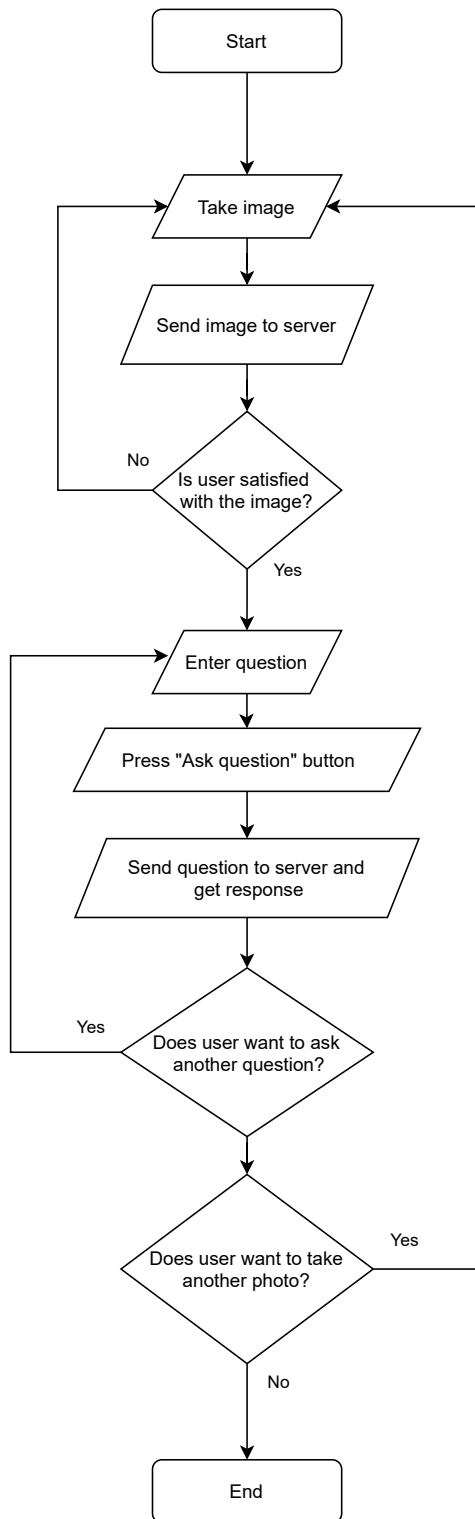


Figure 4.9: Application flow diagram.

## Chapter 5

# VQA versus Image Captioning

Based on the results obtained from participants in the Section 4.3 blind and visually impaired people find Visual Question Answering (VQA) useful and they would appreciate it if their image captioning (IC) smartphone application would also offer VQA. The aim of this chapter is to find out whether the reason why it is not used in practice is the fact that image captioning is better than VQA for similar applications. To obtain this information are performed experiments.

The first Section 5.1 deals with a collection of a custom dataset. This dataset forms a basis for multiple experiments described in the following sections. Then (Section 5.2) are evaluated and compared two different VQA methods on a custom dataset and better performing one is chosen for other experiments (as well as for the application in Chapter 4). The aim of the next Section 5.3 is to generate captions for images in custom dataset. The succeeding Section 5.4 offers an experiment that compare amount of obtained knowledge from VQA and IC. Then is described another experiment (Section 5.5) to find out how well can be VQA used to obtain information about image with participant, which is not able to see that image. Finally (Section 5.6) results obtained in this chapter are discussed.

### 5.1 Dataset Collection

It is believed that to this day that image captioning models are not able to generate superior captions for images to those, created by humans. For this reason was created a custom dataset of 111 images. Due to the small size, the dataset can be annotated manually. All of the images in the dataset are individually chosen from a personal collection. These images consist of diverse scenes and often multiple objects with some of them being hard to recognize. Three captions and one question were created for each image. They were all created by people where the creator of the questions did not know about the captions. Each caption is a single sentence trying to capture the meaning of a corresponding image. These three sets of captions were created by three different people independently. The question is a single sentence asking a question easily answerable by humans but possibly challenging for a machine. An example of images, their captions and a question from the dataset can be seen in Figure 5.1. At the second image can be noticed a significant difference between the generated and the human-created captions. Despite all the human captions mentioning the fact that the house is upside down, the caption generated by the image captioning model does not mention this information.

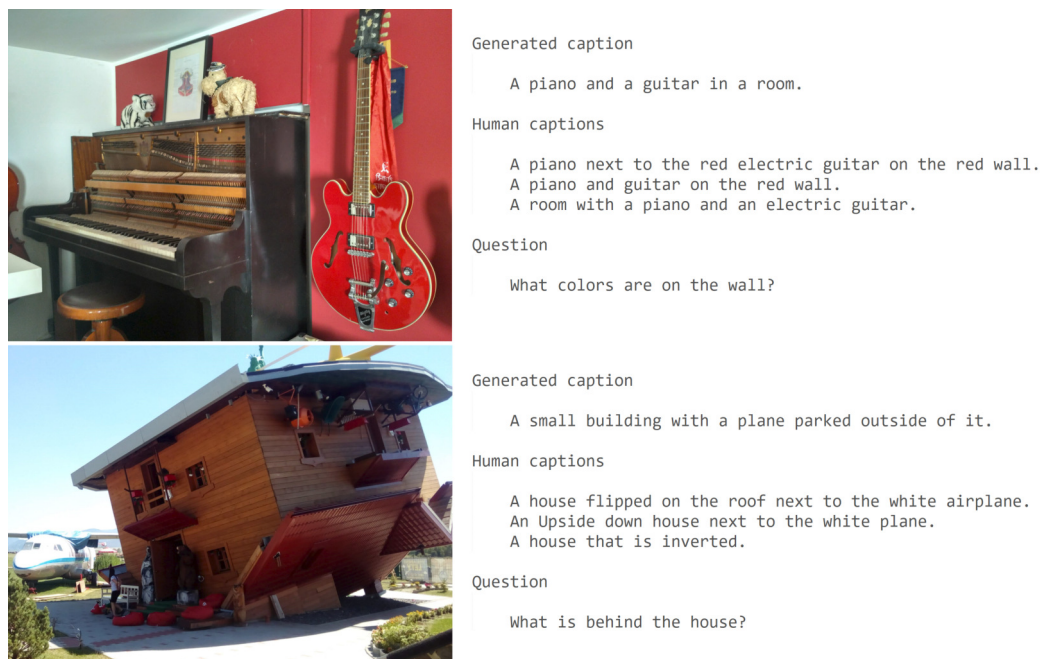


Figure 5.1: Example of images, captions and questions from custom dataset.

## Preparing the Dataset

For the experiments covered later in this chapter another set of captions was created. This set was generated by an image captioning model. I chose the model Oscar (Section 3.2). At the time of creating these captions authors of the model, Oscar reported the state-of-the-art performance for image captioning and also for visual question answering. Another reason why this model was chosen is that the pre-trained checkpoint for this model was publicly available. Since the model was expecting a particular set of data, implementation details required a slight modification<sup>1</sup> for the custom dataset. A file of JavaScript Object Notation (JSON) format consisting of two items for each image was created together with dataset. The first one `img_id` represents image identifier and `img_fn` consists a name with file extension of that image. The process that follows is performed in two steps. The goal of the first step is to extract the image features. The second step is feeding these features as input to the model for specific tasks such as image captioning or visual question answering. The model [1] available on github<sup>2</sup> is used for extracting the image features. The model is based on the Faster R-CNN object detector (Section 2.3.1) implemented in the deep learning framework Caffe [27]. This framework is open source and emphasizes speed and modularity. Even though it is written in C++, the interface is available for Python. Extracted objects and their labels are saved to `tsv` files. Each line of `feature.tsv` consists of extracted features for single image. Based on these files `lineidx` files<sup>3</sup> are created, which are required as input to the Oscar image captioning model. The `lineidx` file differs from `tsv` file by adding size (in bytes) in front of each line separated from rest of the line by tabulator.

<sup>1</sup><https://github.com/microsoft/Oscar/issues/33>

<sup>2</sup><https://github.com/peteanderson80/bottom-up-attention>

<sup>3</sup><https://github.com/microsoft/Oscar/issues/32>

## 5.2 Visual Question Answering

The model Oscar used for image captioning can be also used for visual question answering. To be able to use the model for a different task, it requires to be fine-tuned differently. For VQA the pre-trained model is fine-tuned on the VQAv2 dataset [17]. The JSON file mentioned in the Section 5.1 is modified by adding questions. The input to the VQA model consists of extracted features from all of the images in the custom dataset and a JSON file. The output is a file, which contains numbers representing answers. These numbers are then mapped to *pickle* file `trainval_ans2label.pkl`, which contains all possible trained answers. The answers overall were worse than expected. For this reason, I decided to try another method. The chosen model Pythia (explained in Section 3.2) is available to download already pre-trained.

### Evaluation with score Metrics

Answers provided by VQA models Oscar and Pythia were evaluated by human (Figure 5.2). Each answer was evaluated with a **score** of 1-10 (1 - terrible, 10 - very good). Based on the results it is possible to compare the performance of VQA models.



Question

What is the color of the flowers?

Model	Answer	Score
Oscar	yellow	1
Pythia	red	10



Question

How many bins are there?

Model	Answer	Score
Oscar	2	10
Pythia	1	1



Question

What is this place?

Model	Answer	Score
Oscar	street	3
Pythia	garden	3

Figure 5.2: Examples of images, questions and answers generated with VQA models from custom dataset.



Figure 5.3 presents results of custom metric for VQA. The averages for Pythia and Oscar are 6.95 and 5.40 respectively. Based on these results the performance of Pythia seems to be superior to Oscar, therefore, the Pythia is used for demonstrative application and other experiments.

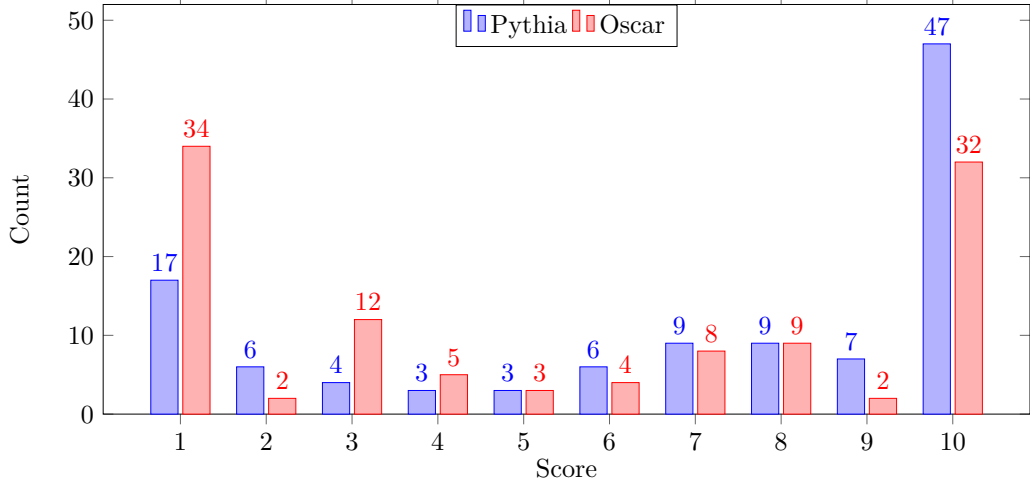


Figure 5.3: Bar graph showing number of 1-10 score for VQA models Pythia and Oscar.

### 5.3 Image Captioning

The model Oscar requires (Section 3.2) fine-tuning so that it can be used for image captioning. The fine-tuning is done in two phases as proposed<sup>4</sup> by authors to improve the score for CIDEr [60] metric. The proposed design was created for GPU with 32GB of memory, but available was just GPU with 12GB. For this reason, the batch size for training was reduced four times, but compensated by setting gradient accumulation steps to four. The first phase is to train to minimize cross-entropy loss. The second is a Self-critical Sequence Training [52] based on reinforcement learning, which should improve image captioning performance.

The scores (achieved at fine-tuning on MS-COCO [39] dataset) from metrics such as BLEU@4<sup>5</sup> (0.366 → 0.404) or CIDEr<sup>6</sup> (1.124 → 1.355) were improved. on the other hand, the captions generated were significantly worse, than before the second phase of fine-tuning. For the custom dataset, all of the sentences were no longer ended with a dot to indicate the end of the sentence. Additionally, 83.78% of the last generated words were either *a* or *the*. This finding was confirmed not only on the custom dataset (Figure 5.4) but also on the validation split of MS-COCO. For this reason, the captions for the custom dataset are generated with Oscar fine-tuned just with cross-entropy.

<sup>4</sup>[https://github.com/microsoft/Oscar/blob/master/MODEL\\_ZOO.md#Image-Captioning-on-COCO](https://github.com/microsoft/Oscar/blob/master/MODEL_ZOO.md#Image-Captioning-on-COCO)

<sup>5</sup>BLEU is an algorithm used in NLP for evaluating the quality of text-based on n-gram overlaps between prediction and references.

<sup>6</sup>CIDEr measures how well a candidate sentence matches the consensus of a set of descriptions. The authors claim, that this metric correlates better with human judgment than BLEU.



Generated caption before CIDEr fine-tuning:

A church with a steeple and a mountain in the background.

Generated caption with CIDEr fine-tuning:

A building with a on a in the on a

Figure 5.4: Example of an image from a custom dataset to demonstrate Oscar image captioning after the first and the second phase of fine-tuning.

### Evaluation with score-correct-generic Metrics

The captions were scored by human using custom metrics called **score-correct-generic**, which is custom created. The metric contains **score** of 1 – 10 (1 - terrible, 10 - very good) for each image, which represents how well does each caption describe the corresponding image. Because the score was not able to cover all the information, another part of this metric was determining if the caption is **correct** at all with values of 0 or 1. Although there is a positive correlation between the high **score** and the metric **correct**, these metrics are not related to each other. Even though some of the captions were labelled as correct, these captions could be missing the main reasoning of the image. Therefore, another metric was added to determine if the captions are too **generic** with values of 0 or 1 (Figure 5.5).



Generated caption

A couple of cars parked next to each other in the snow.

Score	Correct	Generic
7/10	1	1

Figure 5.5: Generated caption for image from custom dataset. The caption is correct, but human would probably mention that the cars have wheel clamps, hence the caption is too generic.

## Results

The averages of results for image captioning are shown in Table 5.1 and distribution of scores is visualised in Figure 5.6. Even though 78% of captions were labelled as correct the overall average score is just 6.1/10. The reason for this ambiguity could be the high overall genericity of captions. Based on these results image captioning is not able to obtain specific details about images.

score	6.1/10
correct	0.78/1.0
generic	0.54/1.0

Table 5.1: The averages of custom metrics for a custom dataset (111 images).

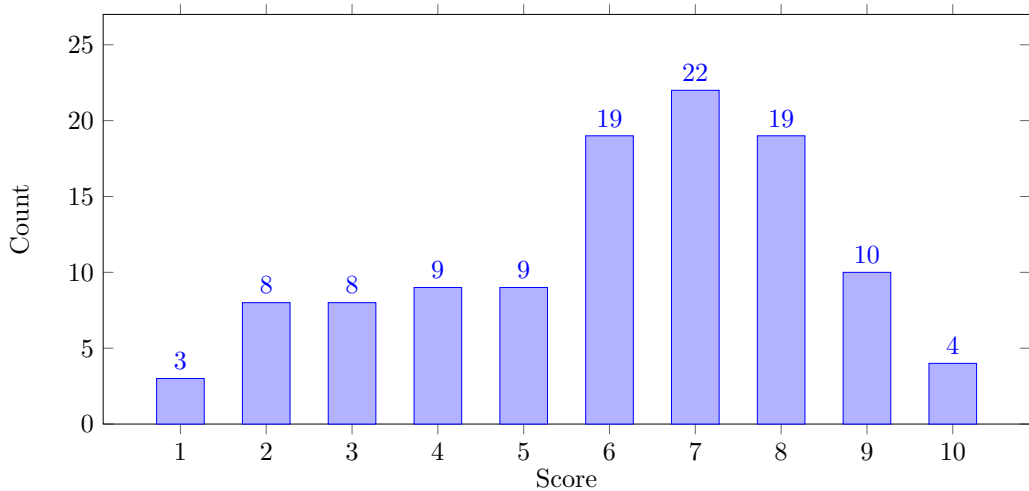


Figure 5.6: Bar graph showing number of 1-10 score for image captioning model Oscar.

## 5.4 Comparison of Obtained Knowledge between VQA and Image Captioning

Hypothesis: *VQA is not used in practice because the image captioning provides enough useful information.*

The purpose of this experiment is to find out if VQA provides more useful information in comparison with image captioning. The participant is presented with a question and a caption (Figure 5.7) for each image without seeing an actual image. The task is to answer this question based only on the provided caption. All the data used in this experiment are from the custom dataset. The participant provides an answer or answers **unknown** if the caption does not provide enough information to be able to answer confidently. This experiment consists of two main parts. The first is based on human-created captions and the second uses captions generated with the Oscar image captioning model. Each of these parts was performed by three different participants. Participants received a different set of human-created captions, but the same set of generated captions. The results of these parts were compared with answers provided by the VQA model Pythia (Section 5.2).

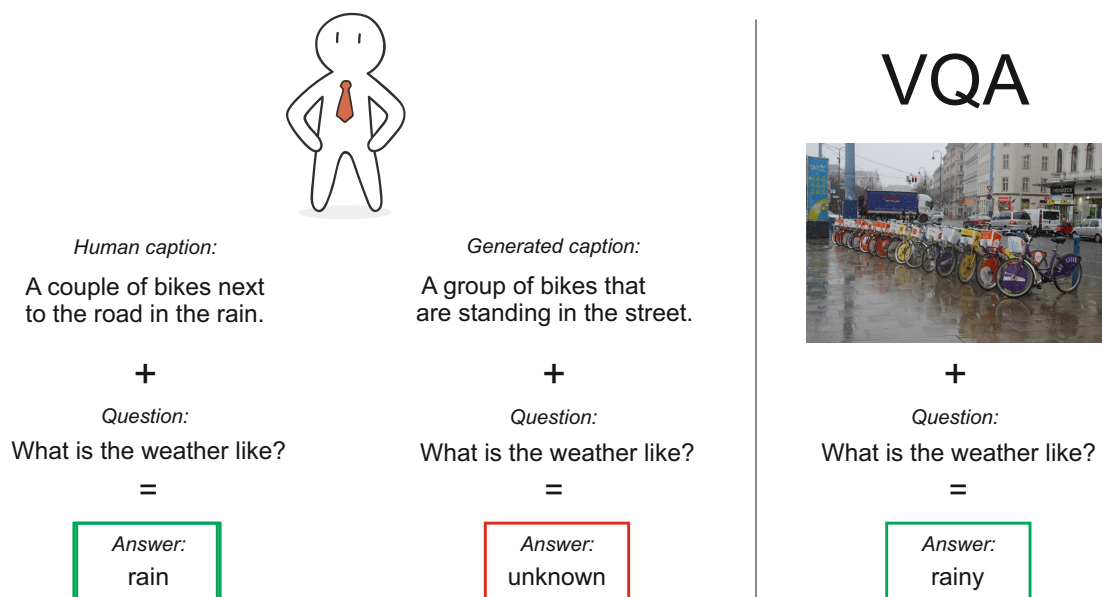


Figure 5.7: Participant is presented with a question and either human created or generated caption from custom dataset. Participant answers **unknown** if caption does not provide enough information. Answers from participants were compared with answers provided by VQA model.

## Evaluation

Since the answers were not chosen from a set of possible answers, there was a significant variety of possible correct answers. Therefore, if the results would be evaluated based on ground truth answers, it would require many sets of ground-truth answers. Based on my manual inspection, almost all of the answers other than **unknown** answered the question correctly or very close to being correct, thus I decided to label as correct all answers other than **unknown**. In other words, the results are based on the participant being able to answer or not. The answers obtained with the VQA model as was described in the previous section were labelled based on their score. The threshold for the answer to be labelled as correct was set to score  $\geq 8$  to also include answers very close to being exactly correct.

## Results

The average portions of correct answers for human and generated caption were 58.1% and 46.2% respectively (Table 5.2). In comparison, 69.9% of answers generated with the VQA model was labelled as correct. These results suggest that VQA is able to perform superior performance to image captioning regarding specific details about images. Based on the results was calculated  $\kappa$  statistic measure as described in Section 2.5. Value of  $\kappa$  for human caption is 0.72, for generated captions it is 0.86. Finally for all the captions combined  $\kappa = 0.74$ , which could be interpreted as substantial agreement.

	Human caption [%]	Generated caption [%]
Person 1	65.5	48.8
Person 2	55.5	43.3
Person 3	53.3	46.6
Average	58.1	46.2

Table 5.2: The portion of answers labeled as correct for each of the participants.

## 5.5 Reasoning over Images with VQA

Hypothesis: *VQA can not be used without seeing the actual image.*

The aim of this experiment is to find out if the reason why the VQA is not really used in real-world applications is its inability to provide useful information on its own. The testing with users showed that they would appreciate a combination of VQA and image captioning. The results should show how well can be understood image from information obtained with VQA without seeing the actual image (Figure 5.8).



Figure 5.8: An example of image, which is hidden to the participant.

A similar application as was used for testing with users is used for this experiment. This time the application does not show the actual image. Apart from the application participant is presented with a simple script. Upon starting the script user is presented with 5 different captions, where only one of them is correct. The other four captions are randomly selected from the rest of the dataset.

1. A tree next to the wooden cabin with the red roof.
2. Platform at the train station with blue sky in the background.
3. A tent covering sound equipment next to the statue.
4. People browsing Christmas sales on the street.
5. A red traffic light with trees in the background.

The goal is to select the most appropriate caption for the image they are not able to see. To reach this goal each participant is allowed to ask at least one but no more than three questions using VQA (Table 5.3). Based on the captions above participant asked the following questions. Upon manual inspection, I noticed the first one was rather generic to try to narrow down the selection of captions. The next question focuses on determining whether any humans are present. The last question is much more specific to confirm or deny the third caption.

Question	Answer
What is in the foreground?	sign
How many people are there?	0
Is there any statue?	no

Table 5.3: An example of questions and answers from single participant for an example shown in Figure 5.8.

After questions are answered participant must select one of the captions. Then a new set of 5 captions is generated again for a new image and the whole process is repeated for all the images. This experiment is performed by three people. Each of them receives captions from a different set of captions in the custom dataset. All questions and answers from the VQA model were stored in the log file.

## Results

The results (Table 5.4) show a positive correlation between portions of yes or no answers and the higher average of correctly answered. Between the participants is an apparent difference in approaches. Upon manual analysis, it was found that one of the participants significantly differs from others by asking more complex questions, which does not result in yes or no answers. Based on the results, this approach leads to worse performance. From correct answers  $\kappa$  Fleiss (Section 2.5) was calculated. The  $\kappa$  value 0.62 could be interpreted as substantial agreement.

	Correctly answered [%]	Yes/no answer to question [%]	Questions per image	Question length
Person 1	77.48	75.17	2.68	3.98
Person 2	70.27	67.42	2.38	4.61
Person 3	64.86	35.34	2.55	4.29
Average	70.87	59.31	2.54	4.29

Table 5.4: The portions of correctly answered, yes or no answers, average number of questions out of 3 allowed and an average number of words per question asked.



## 5.6 Summarizing Discussion

The issue with the second phase of fine-tuning of the image captioning model (Section 5.3) could be caused by optimizing towards the proxy metrics which were never meant to be optimized for. In fact, these metrics were proposed as a rough approximate to the human evaluation, as it was found they correlate well with it. Solving this issue requires further research.

Captions generated by the image captioning model are often able to capture objects and some of their relations correctly or in a similar meaning. Nevertheless, these captions sometimes suffer from missing the precise meaning of the given image and focus more on other detail of such an image, which in some cases might be insufficient. This problem could be partly solved by using image captioning together with VQA, which is able to focus on specific details of the image. On the other hand, this combination adds complexity. Image captioning requires no additional input other than an image, while for VQA it is necessary to create a question. To obtain any useful knowledge, this question needs to be well-formulated.

All these experiments were evaluated with custom metrics. Generated captions could be compared with human captions by using metrics such as BLEU, but these metrics also suffer from significant drawbacks. BLEU does not consider meaning, sentence structure or synonyms. VQA answers could be compared with ground truth answers, but then again, synonyms or similar words would not receive any score whatsoever. Used metrics are subjective, but for the dataset smaller in size, human judgements are preferred.

Although questions for the experiment (Section 5.4) were the same for all of the captions, the agreement  $\kappa$  for human captions is significantly lower (human 0.72, generated 0.86). The reason is that generated captions used were the same for all participants, unlike the human captions, where each of the participants received a different set of captions. For the other experiment (Section 5.5) the value of  $\kappa = 0.62$  is caused by the selection of captions. Even though the correct captions were the same for all participants, the other 4 captions, which were wrong, were selected at random. For this reason, the captions in this experiment were different for different participants.

Results for the experiment described in Section 5.4 disprove the presented hypothesis (*VQA is not used in practice because the image captioning provides enough useful information.*). Based on this experiment, the VQA model is able to provide more knowledge about the details of a given image, than a caption generated by an image captioning model. The VQA model outperformed generated captions and also captions created by humans.

The other experiment's (Section 5.5) hypothesis (*VQA can not be used without seeing the actual image.*) is not supported by results. On average 70.87% of captions were chosen correctly. This experiment took more than two hours for each participant to perform. Although VQA is able to provide useful information even if the user does not see the image, formulating and asking questions takes a lot more time than using only image captioning.

## Chapter 6

# Discussion and Conclusion

The objective of this thesis was to narrow the gap between research and practice. When studying the actual usage of VQA in practice, I found applications that use image captioning (IC), but I struggled to find ones that would use VQA. To find a use case for VQA, I contacted the blind and visually impaired (BVI) community. I studied how they use technology, and then I created a VQA demonstrative app and made it accessible to BVI. I let them test the app, gathered knowledge with a questionnaire and created a smartphone app. Based on the answers provided, the participants consider VQA to be useful. 80% of them would appreciate it if their IC app also offered VQA. Some think that the focus of the app should be specified either for implementation in a browser for online shopping or as a smartphone app that could be used for orientation and localization or choosing clothes. Although the overall rating was positive, these results could be biased for the following reason. I communicated directly with about a dozen people from the BVI community. Some of these people were willing to redistribute the app and questionnaire to their friends in the community. For this reason, I do not have exact numbers on how many people tried the application and how many of them filled out the questionnaire. Thus, a positive evaluation could be caused simply by a situation where people who do not like the application are not willing to fill in the questionnaire at all. I tried to find out if IC is used more often than VQA because it is better. For this purpose, I created a novel dataset and designed unique experiments. To summarize the results, VQA works better in terms of specific image details. The disadvantage, on the other hand, is that, unlike IC, which does not require any input other than an image, a well-formulated question needs to be added to VQA. While preparing the experiments, I encountered a problem with fine-tuning the IC model. To fine-tune the IC model, the authors used reinforcement learning to improve the performance of metrics used for IC. Although metrics-based performance improved, the generated captions were worse. Instead of refining models to achieve better scores on these metrics, research should focus more on usefulness of presented approaches on the downstream tasks or in practice. The results of this work open up new possibilities for further research. One of the possibilities for future work is a browser extension for BVI. This extension could offer VQA for any image encountered while browsing websites. Although 20% of respondents think it could be used for browsing social networks, 45% of participants would appreciate such an application for online shopping. Products online often rely on images and description does not say a lot about the visual aspect of an item. This use case is even more important in times when a lot of products are simply impossible to buy in person.

# Bibliography

- [1] ANDERSON, P., HE, X., BUEHLER, C., TENEY, D., JOHNSON, M. et al. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, p. 6077–6086. DOI: 10.1109/CVPR.2018.00636.
- [2] ANTOL, S., AGRAWAL, A., LU, J., MITCHELL, M., BATRA, D. et al. VQA: Visual Question Answering. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, p. 2425–2433. DOI: 10.1109/ICCV.2015.279.
- [3] CHO, K., MERRIËNBOER, B. van, GULCEHRE, C., BAHDANAU, D., BOUGARES, F. et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, October 2014, p. 1724–1734. DOI: 10.3115/v1/D14-1179.
- [4] CHUNG, J., GULCEHRE, C., CHO, K. and BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: *NIPS 2014 Workshop on Deep Learning, December*. 2014.
- [5] DENG, J., DONG, W., SOCHER, R., LI, L., KAI LI et al. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, p. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [6] DEVLIN, J., CHANG, M.-W., LEE, K. and TOUTANOVA, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: . June 2019.
- [7] EVERINGHAM, M., ESLAMI, S. M., GOOL, L., WILLIAMS, C. K., WINN, J. et al. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vision*. USA: Kluwer Academic Publishers. january 2015, vol. 111, no. 1, p. 98–136. DOI: 10.1007/s11263-014-0733-5. ISSN 0920-5691.
- [8] FANG, H., GUPTA, S., IANDOLA, F., SRIVASTAVA, R., DENG, I. et al. From captions to visual concepts and back. In: . June 2015, p. 1473–1482. DOI: 10.1109/CVPR.2015.7298754.
- [9] FISCHLER, M. A. and ELSCHLAGER, R. A. The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*. 1973, C-22, no. 1, p. 67–92. DOI: 10.1109/T-C.1973.223602.
- [10] FLEISS, J. Measuring nominal scale agreement among many raters. *Psychological bulletin*. November 1971, vol. 76, no. 5, p. 378–382. DOI: 10.1037/h0031619. ISSN 0033-2909.

- [11] GAUNT, R., PICKETT, A. and REINERT, G. Chi-square approximation by Stein’s method with application to Pearson’s statistic. *The Annals of Applied Probability*. may 2016, vol. 27. DOI: 10.1214/16-AAP1213.
- [12] GIRSHICK, R. Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, p. 1440–1448. DOI: 10.1109/ICCV.2015.169.
- [13] GIRSHICK, R., DONAHUE, J., DARRELL, T. and MALIK, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, p. 580–587. DOI: 10.1109/CVPR.2014.81.
- [14] GIRSHICK, R., RADOSAVOVIC, I., GKIOXARI, G., DOLLÁR, P. and HE, K. *Detectron* [<https://github.com/facebookresearch/detectron>]. 2018.
- [15] GOLDBERG, Y. *Neural Network Methods for Natural Language Processing*. San Rafael, CA: Morgan & Claypool, 2017. Synthesis Lectures on Human Language Technologies. ISBN 978-1-62705-298-6.
- [16] GOODFELLOW, I., BENGIO, Y. and COURVILLE, A. *Deep Learning*. MIT Press, 2016 [cit. 2021-01-22]. <http://www.deeplearningbook.org>.
- [17] GOYAL, Y., KHOT, T., AGRAWAL, A., SUMMERS STAY, D., BATRA, D. et al. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *International Journal of Computer Vision*. april 2019, vol. 127. DOI: 10.1007/s11263-018-1116-0.
- [18] GRAND, G. *Learning Interpretable and Bias-Free Models for Visual Question Answering*. 2019. Bachelor’s Thesis. Harvard College.
- [19] GU, J., WANG, Z., KUEN, J., MA, L., SHAHROUDY, A. et al. Recent advances in convolutional neural networks. *Pattern Recognition*. 2018, vol. 77, p. 354–377. DOI: <https://doi.org/10.1016/j.patcog.2017.10.013>. ISSN 0031-3203.
- [20] HE, K., ZHANG, X., REN, S. and SUN, J. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, p. 770–778. DOI: 10.1109/CVPR.2016.90.
- [21] HEITZ, G., ELIDAN, G., PACKER, B. and KOLLER, D. Shape-Based Object Localization for Descriptive Classification. *International Journal of Computer Vision*. 2009, vol. 84, p. 40–62.
- [22] HOCHREITER, S. and SCHMIDHUBER, J. Long short-term memory. *Neural computation*. MIT Press. 1997, vol. 9, no. 8, p. 1735–1780.
- [23] HOWARD, J. and RUDER, S. Universal Language Model Fine-tuning for Text Classification. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, p. 328–339. DOI: 10.18653/v1/P18-1031.
- [24] HUBEL, D. H. and WIESEL, T. N. Receptive Fields of Single Neurons in the Cat’s Striate Cortex. *Journal of Physiology*. 1959, vol. 148, p. 574–591.

- [25] HUDSON, D. A. and MANNING, C. D. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, p. 6693–6702. DOI: 10.1109/CVPR.2019.00686.
- [26] IOFFE, S. and SZEGEDY, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: BACH, F. and BLEI, D., ed. *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France: PMLR, 07–09 Jul 2015, vol. 37, p. 448–456.
- [27] JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J. et al. Caffe: Convolutional Architecture for Fast Feature Embedding. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*. june 2014. DOI: 10.1145/2647868.2654889.
- [28] JIANG, H., MISRA, I., ROHRBACH, M., LEARNED-MILLER, E. and CHEN, X. In Defense of Grid Features for Visual Question Answering. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, p. 10264–10273. DOI: 10.1109/CVPR42600.2020.01028.
- [29] JIANG, Y., NATARAJAN, V., CHEN, X., ROHRBACH, M., BATRA, D. et al. Pythia v0.1: the Winning Entry to the VQA Challenge 2018. *CoRR*. 2018, abs/1807.09956.
- [30] KAFLE, K. and KANAN, C. Visual Question Answering: Datasets, Algorithms, and Future Challenges. *Computer Vision and Image Understanding*. october 2016, vol. 163. DOI: 10.1016/j.cviu.2017.06.005.
- [31] KARPATHY, A. and FEI-FEI, L. Deep visual-semantic alignments for generating image descriptions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 3128–3137. DOI: 10.1109/CVPR.2015.7298932.
- [32] KRISHNA, R., LI, F.-F. and XU, D. *Convolutional Neural Networks (CNNs / ConvNets)*. 2020 [cit. 2021-01-13]. Available at: <https://cs231n.github.io/convolutional-networks/>.
- [33] KRISHNA, R., ZHU, Y., GROTH, O., JOHNSON, J., HATA, K. et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vision*. USA: Kluwer Academic Publishers. may 2017, vol. 123, no. 1, p. 32–73. DOI: 10.1007/s11263-016-0981-7. ISSN 0920-5691.
- [34] LAN, Z., CHEN, M., GOODMAN, S., GIMPEL, K., SHARMA, P. et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In: *International Conference on Learning Representations*. 2020.
- [35] LANDIS, J. R. and KOCH, G. G. The Measurement of Observer Agreement for Categorical Data. *Biometrics*. [Wiley, International Biometric Society]. 1977, vol. 33, no. 1, p. 159–174. ISSN 0006341X, 15410420.
- [36] LAWRENCE, S., GILES, C., TSOI, A. and BACK, A. Face Recognition: A Convolutional Neural Network Approach. *Neural Networks, IEEE Transactions on*. february 1997, vol. 8, p. 98 – 113. DOI: 10.1109/72.554195.
- [37] LEBRET, R., PINHEIRO, P. H. O. and COLLOBERT, R. Simple Image Description Generator via a Linear Phrase-Based Approach. *CoRR*. 2015, abs/1412.8419.

- [38] LI, X., YIN, X., LI, C., HU, X., ZHANG, P. et al. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. *European Conference on Computer Vision 2020*. 2020.
- [39] LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P. et al. Microsoft COCO: Common Objects in Context. In: FLEET, D., PAJDLA, T., SCHIELE, B. and TUYTELAARS, T., ed. *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, p. 740–755.
- [40] LOWE, D. G. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999, vol. 2, p. 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.
- [41] MALINOWSKI, M. and FRITZ, M. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In: GHARAMANI, Z., WELLING, M., CORTES, C., LAWRENCE, N. and WEINBERGER, K. Q., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, vol. 27.
- [42] MARR, D. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982. ISBN 0716715678.
- [43] MIKOLOV, T., CHEN, K., CORRADO, G. and DEAN, J. Efficient Estimation of Word Representations in Vector Space. In: BENGIO, Y. and LECUN, Y., ed. *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. 2013.
- [44] NORCLIFFE BROWN, W., VAPEIAS, E. and PARISOT, S. Learning Conditioned Graph Structures for Interpretable Visual Question Answering. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 8344–8353. NIPS’18.
- [45] OLAH, C. *Understanding LSTM Networks*. 2015 [cit. 2021-01-15]. Available at: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [46] OTTER, D. W., MEDINA, J. R. and KALITA, J. K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*. 2021, vol. 32, no. 2, p. 604–624. DOI: 10.1109/TNNLS.2020.2979670.
- [47] PAPERT, S. *The Summer Vision Project*. Massachusetts Institute of Technology, Project MAC, 1966. AI memo. Available at: <https://books.google.cz/books?id=q0h7NwAACAAJ>.
- [48] PASCANU, R., MIKOLOV, T. and BENGIO, Y. On the difficulty of training recurrent neural networks. In: DASGUPTA, S. and MCALLESTER, D., ed. *Proceedings of the 30th International Conference on Machine Learning*. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, vol. 28, no. 3, p. 1310–1318.
- [49] PETERS, M., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C. et al. Deep Contextualized Word Representations. In: *Proceedings of the 2018 Conference of the NAACL: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: ACL, June 2018, p. 2227–2237. DOI: 10.18653/v1/N18-1202.



- [50] REN, M., KIROS, R. and ZEMEL, R. Exploring Models and Data for Image Question Answering. In: CORTES, C., LAWRENCE, N., LEE, D., SUGIYAMA, M. and GARNETT, R., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015, vol. 28.
- [51] REN, S., HE, K., GIRSHICK, R. and SUN, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: CORTES, C., LAWRENCE, N., LEE, D., SUGIYAMA, M. and GARNETT, R., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015, vol. 28.
- [52] RENNIE, S. J., MARCHERET, E., MROUEH, Y., ROSS, J. and GOEL, V. Self-Critical Sequence Training for Image Captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, p. 1179–1195.
- [53] SHI, J. and MALIK, J. Normalized Cuts and Image Segmentation. In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. USA: IEEE Computer Society, 1997, p. 731. CVPR '97. ISBN 0818678224.
- [54] SILBERMAN, N., HOIEM, D., KOHLI, P. and FERGUS, R. Indoor Segmentation and Support Inference from RGBD Images. In: FITZGIBBON, A., LAZEBNIK, S., PERONA, P., SATO, Y. and SCHMID, C., ed. *Computer Vision – ECCV 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, p. 746–760. ISBN 978-3-642-33715-4.
- [55] SIMONYAN, K. and ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: BENGIO, Y. and LECUN, Y., ed. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.
- [56] SINGH, A., GOSWAMI, V., NATARAJAN, V., JIANG, Y., CHEN, X. et al. *MMF: A multimodal framework for vision and language research* [<https://github.com/facebookresearch/mmf>]. 2020.
- [57] SUNNY KATIYAR, M. S. W. A Survey On Visual Questioning Answering : Datasets, Approaches And Models. *International Journal of Scientific & Technology Research*. january 2020, vol. 9, p. 5. ISSN 2277-8616.
- [58] SUTSKEVER, I., VINYALS, O. and LE, Q. V. Sequence to Sequence Learning with Neural Networks. In: GHAHRAMANI, Z., WELLING, M., CORTES, C., LAWRENCE, N. and WEINBERGER, K. Q., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, vol. 27.
- [59] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L. et al. Attention is All you Need. In: GUYON, I., LUXBURG, U. V., BENGIO, S., WALLACH, H., FERGUS, R. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, vol. 30.
- [60] VEDANTAM, R., ZITNICK, C. L. and PARIKH, D. CIDEr: Consensus-based image description evaluation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 4566–4575. DOI: 10.1109/CVPR.2015.7299087.
- [61] VINYALS, O., TOSHEV, A., BENGIO, S. and ERHAN, D. Show and tell: A neural image caption generator. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, p. 3156–3164. DOI: 10.1109/CVPR.2015.7298935.

- [62] VIOLA, P. and JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001, vol. 1, p. I–I. DOI: 10.1109/CVPR.2001.990517.
- [63] WU, Q., TENNEY, D., WANG, P., SHEN, C., DICK, A. et al. Visual Question Answering: A Survey of Methods and Datasets. *Computer Vision and Image Understanding*. july 2016, vol. 163. DOI: 10.1016/j.cviu.2017.05.001.
- [64] WU, Y., SCHUSTER, M., CHEN, Z., LE, Q. V., NOROUZI, M. et al. *Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation* [1609.08144]. ArXiv. 2016.
- [65] XIE, S., GIRSHICK, R., DOLLÁR, P., TU, Z. and HE, K. Aggregated Residual Transformations for Deep Neural Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, p. 5987–5995. DOI: 10.1109/CVPR.2017.634.
- [66] XU, J., SUN, X., ZHANG, Z., ZHAO, G. and LIN, J. Understanding and Improving Layer Normalization. In: WALLACH, H., LAROCHELLE, H., BEYGEZIMER, A., ALCHÉ BUC, F. d’, FOX, E. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019, vol. 32.
- [67] XU, K., BA, J., KIROS, R., CHO, K., COURVILLE, A. et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: BACH, F. and BLEI, D., ed. *Proceedings of the 32nd ICML*. Lille, France: PMLR, 07–09 Jul 2015, vol. 37, p. 2048–2057.
- [68] YOU, Q., JIN, H., WANG, Z., FANG, C. and LUO, J. Image Captioning with Semantic Attention. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, p. 4651–4659. DOI: 10.1109/CVPR.2016.503.
- [69] YU, L., PARK, E., BERG, A. and BERG, T. Visual Madlibs: Fill in the blank Image Generation and Question Answering. may 2015. DOI: 10.1109/ICCV.2015.283.
- [70] ZHANG, A., LIPTON, Z. C., LI, M. and SMOLA, A. J. *Dive into Deep Learning*. [cit. 2021-01-19]. <https://d2l.ai>.
- [71] ZHANG, J., HE, T., SRA, S. and JADBABAIE, A. Why Gradient Clipping Accelerates Training: A Theoretical Justification for Adaptivity. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30*. 2020.
- [72] ZHANG, P., LI, X., HU, X., YANG, J., ZHANG, L. et al. VinVL: Making Visual Representations Matter in Vision-Language Models. *CVPR 2021*. 2021.
- [73] ZHU, Y., GROTH, O., BERNSTEIN, M. S. and FEI FEI, L. Visual7W: Grounded Question Answering in Images. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, p. 4995–5004.