People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
AMO University of Bouira
Faculty of Sciences and Applied Sciences
Computer Science Department

Master Thesis

in Computer Science

Speciality: *Computer Systems Engineering -GSI-*

Theme

Visual Question Answering using Deep Learning

Supervised By

- Aid Aicha

Realized By

- Moulai Azeddine

2023/2024

Acknowledgment

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout our research work to complete the research successfully.
We would like to express our deep and sincere gratitude to our research supervisors, **Dr. AID Aicha** , **Dr. DJOUABRI Abderezzak** , **Dr. OUAKAS Noureddine** and **Mr.Haddouche Ahcene** for giving us the opportunity to do this research and providing invaluable guidance throughout the whole research. Their dynamism, vision, sincerity and motivation have deeply inspired us.

They have taught us the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under their guidance.

We are extremely grateful for what they have offered us. We would also like to thank them for their friendship, empathy, and great sense of humor.

We are very much thankful to our friends for their love, understanding, prayers and continuing support to complete this research work. Also, we express our thanks to our sisters for their support and valuable prayers.

We would like to say thanks to our teachers throughout our years of study in the University of Bouira, for their constant
encouragement.
we express our special thanks to them for their genuine support throughout this research work.

Finally, our thanks go to all the people who have supported us to complete the research work directly or indirectly.

Dedication

TO My mother

A strong and gentle soul, words just can't describe her beautiful soul. I can never thank her enough for being by my side through all my life

TO the memory of My father

Who always believed in my ability to be successful in my life. You are gone but your belief in me has made this journey possible.

TO the memory of my brother Abdelhamid

Whose memory continues to inspire and motivate me every day.

TO the memory of my sister Nadjia

Whose love and kindness will always remain in my heart.

My beloved brothers and sisters, relatives , mentor, classmates and friends for bearing me and supporting me and the huge moral support.

I want to extend a special thank you to my classmates. Love and support from all of you have been invaluable. You all have been my second family, standing by me through thick and thin: Farouk, Oualid, Haithem, Fatima, Houda, and Yasmine.

My classmates who became my second family, for being by my side through thick and thin, who continually provide their moral, spiritual and emotional support.

To only those who can understand this code: 11011 11111 10101 11111.

Moulai Azeddine .

Abstract

Motivation

Visual Question Answering (VQA) in the field of artificial intelligence (AI) integrates computer vision and natural language processing to develop systems capable of answering questions based on visual content. The rapid advancements in AI research have significantly expanded the complexity and variety of data available for training VQA models. This growing volume of visual data and the diverse nature of questions posed necessitate sophisticated techniques to achieve high performance. VQA systems have immense potential in real-world applications such as assisting visually impaired individuals, enhancing human-computer interaction, and improving automated customer support. However, developing effective VQA systems remains challenging due to the need for precise understanding and integration of visual and textual information. Transformer-based architectures, with their attention mechanisms, have revolutionized natural language processing and are now making significant inroads into computer vision. These models excel at capturing dependencies and relationships within data, making them well-suited for tasks that require an understanding of both images and text.

Objectives

This study aims to explore the application of transformer models in VQA, focusing on leveraging deep learning techniques to improve system performance. The main objective is to enhance the ability of VQA systems to answer questions accurately and efficiently by fine-tuning pre-trained transformer models with domain-specific data. Additionally, we conducted a comparative study between several state-of-the-art transformer models to evaluate their performance and identify the most effective model.

Results

Our research delves into the architecture of transformer models, their training processes, and the critical datasets for their development. Through fine-tuning pre-trained transformer models on the VizWiz dataset, our approach demonstrated an accuracy of 80%, surpassing related works. This study confirms the potential of transformer-based archi-

tectures to significantly enhance VQA systems, making them more accurate and efficient for real-world applications.

Keywords

Visual Question Answering, Transformer Models, Deep Learning, Computer Vision, Natural Language Processing, VizWiz Dataset, AI, Human-Computer Interaction

Résumé

Motivation

Le Visual Question Answering (VQA) dans le domaine de l'intelligence artificielle (IA) intègre la vision par ordinateur et le traitement du langage naturel pour développer des systèmes capables de répondre à des questions basées sur du contenu visuel. Les progrès rapides de la recherche en IA ont considérablement élargi la complexité et la variété des données disponibles pour l'entraînement des modèles VQA. Ce volume croissant de données visuelles et la nature diverse des questions posées nécessitent des techniques sophistiquées pour atteindre des performances élevées. Les systèmes VQA ont un potentiel immense dans les applications réelles telles que l'assistance aux personnes malvoyantes, l'amélioration de l'interaction homme-machine et l'amélioration du support client automatisé. Cependant, développer des systèmes VQA efficaces reste un défi en raison de la nécessité d'une compréhension et d'une intégration précises des informations visuelles et textuelles. Les architectures basées sur les transformateurs, avec leurs mécanismes d'attention, ont révolutionné le traitement du langage naturel et font désormais des avancées significatives dans la vision par ordinateur. Ces modèles excellent à capturer les dépendances et les relations au sein des données, ce qui les rend bien adaptés aux tâches nécessitant une compréhension des images et du texte.

Objectifs

Ce travail vise à explorer l'application des modèles de transformateurs dans le VQA, en se concentrant sur l'utilisation des techniques d'apprentissage profond pour améliorer les performances du système. L'objectif principal est d'améliorer la capacité des systèmes VQA à répondre aux questions de manière précise et efficace en ajustant des modèles de transformateurs pré-entraînés avec des données spécifiques au domaine. De plus, nous avons mené une étude comparative entre plusieurs modèles de transformateurs de pointe pour évaluer leurs performances et identifier le modèle le plus efficace.

Résultats

Notre recherche examine l'architecture des modèles de transformateurs, leurs processus d'entraînement et les ensembles de données critiques pour leur développement. En ajustant des modèles de transformateurs pré-entraînés sur le jeu de données VizWiz, notre approche a démontré une précision de 80%, surpassant les travaux connexes. Cette étude confirme le potentiel des architectures basées sur les transformateurs pour améliorer significativement les systèmes VQA, les rendant plus précis et efficaces pour les applications réelles.

Mots-Clés

Visual Question Answering, Modèles de Transformateurs, Apprentissage Profond, Vision par Ordinateur, Traitement du Langage Naturel, Jeu de Données VizWiz, IA, Interaction Homme-Machine

Contents

List of Figures

List of Tables

Abreviations List

AI	Artificial Intelligence
AR	Augmented Reality
BLIP	Bootstrapping Language-Image Pre-training
CLEVR	Compositional Language and Elementary Visual Reasoning
CLIP	Contrastive Language-Image Pre-training
CNNs	Convolutional Neural Networks
CV	Computer Vision
LSTM	Long Short Term Memory
Mask-R-CNN	Mask Region-based Convolutional Neural Network
MRIs	Magnetic Resonance Imaging
MS COCO	Microsoft Common Objects in Context
NLP	Natural Language Processing
OCR	Optical Character Recognition
PaliGemma	Pali: A jointly-scaled multilingual language-image model
RNN	Recurrent Neural Networks
ViLT	Vision-and-Language Transformer
ViT	Vision Transformer
VizWiz	A dataset designed for visually impaired users
VLP	Vision-Language Pre-training
VQA	Visual Question Answering
VQA v2.0	Visual Question Answering version 2.0

General Introduction:

Every year, the field of Visual Question Answering (VQA) witnesses significant advancements, reflecting the rapid growth in artificial intelligence research. VQA is a challenging interdisciplinary domain that integrates computer vision and natural language processing, aiming to develop systems capable of answering questions based on visual content. As this field evolves, the complexity and variety of data available for training these models also expand, making it imperative to utilize sophisticated techniques to achieve high performance.

The increasing volume of visual data, coupled with the diverse and intricate nature of questions posed, necessitates advanced systems that can accurately interpret and respond to these queries. VQA systems hold immense potential in various real-world applications, such as assisting visually impaired individuals, enhancing human-computer interaction, and improving automated customer support. However, the development of effective VQA systems is challenging due to the need for precise understanding and integration of both visual and textual information

One promising approach to address these challenges is the application of transformer-based architectures, which have revolutionized natural language processing and are now making significant inroads into computer vision. Transformers, with their attention mechanisms, excel at capturing dependencies and relationships within data, making them well-suited for tasks that require an understanding of both images and text.

In this study, we aim to explore the application of transformer models in VQA, focusing on leveraging deep learning techniques to improve system performance. We will delve into the architecture of these models, their training processes, and the datasets that are critical for their development. By fine-tuning pre-trained transformer models with

domain-specific data, we can enhance their ability to answer questions accurately and efficiently.

This document is structured into three main sections:

Chapter One provides a detailed overview of VQA, including the evolution of VQA systems, their architecture, motivations for their development, and their applications. This chapter also reviews key datasets used in the field and discusses various methods and techniques employed in VQA, from foundational approaches to cutting-edge advancements. The evaluation metrics used to assess VQA systems are also covered.

Chapter Two focuses on the transformer architecture, a groundbreaking framework in deep learning. It begins with an introduction to neural networks for natural language processing and computer vision, followed by an in-depth exploration of the attention mechanism and transformer models. The chapter highlights transformer-based models designed for vision-and-language tasks and examines related works in the field.

Chapter Three presents a comparative study of different VQA models, detailing the datasets used, preprocessing methods, hyperparameter tuning, and fine-tuning processes. The chapter also discusses the experimental setup, performance metrics, and challenges encountered. A critical analysis of the results provides insights into the strengths and weaknesses of various approaches.

Through this comprehensive study, we aim to contribute to the growing body of knowledge in VQA, offering valuable insights and practical guidance for researchers and practitioners in the field.

Chapter 1

Visual Question Answering

1.1 Introduction

Visual Question Answering stands as a foundational task within the realm of vision-and-language research, garnering significant interest from diverse artificial intelligence communities such as Computer Vision (CV) and Natural Language Processing (NLP). This task serves as a bridge between CV and NLP, fostering research collaboration and pushing the boundaries of both fields. In its typical form, VQA involves presenting a model with an image accompanied by a textual question related to the image. The model is then tasked with accurately determining and expressing the answer in a concise manner, often using a few words or a short phrase. VQA exhibits various forms, including binary (yes or no answers) and multiple-choice scenarios, where the model must choose from a set of candidate answers [8].

In the following sections of this chapter, we will offer a comprehensive overview of VQA systems, covering aspects like their definitions and real world applications, motivations and goal, methods and techniques of VQA models, and metrics for evaluation. Additionally, we will explore datasets have been curated for training and evaluating VQA models. These datasets typically consist of images, associated questions in natural language, and corresponding ground-truth answers.

1.2 General overview of VQA

1.2.1 General overview

Over the last decade, progress in deep learning systems has resulted in significant breakthroughs in understanding both visual and textual information. This progress has empowered AI models to reach a level where they can compete with human performance in these areas. While humans have traditionally excelled in comprehending images, AI has lagged behind, often producing answers that are considered mediocre and simplistic. However, distinguishing between responses generated by humans and those produced by AI has become increasingly challenging recently. This is evident in the fact that the specific line under consideration has been crafted by an AI system. VQA presents a challenge within the AI domain, drawing heavily from the principles and methodologies of both Computer Vision (CV) and Natural Language Processing (NLP). As VQA has progressed, it has developed its own distinct identity with unique characteristics and subtleties [9]. Figure 1.1 illustrates Sample from VizWiz Dataset: Image, Question, and Answer from Visually Impaired Users [1].

Visual Question Answering is a task that involves answering questions about images. It requires understanding both the visual content of an image and the textual content of a question to provide an accurate natural language answer. VQA task can be summarized as follows:

- Input: An image and a question about the image (e.g., "What color is the car?").
- Output: A natural language answer (e.g., "Blue").

1.2.2 VQA System Architecture

VQA systems need to understand the content of the image, recognize objects, infer relationships, and comprehend the meaning of the question to generate an accurate answer. Therefore, the architecture of a VQA system begins with the extraction of high-level features from input images. Simultaneously, textual questions undergo processing, involving techniques like word embedding to capture semantic meaning. These visual and textual features are then fused to create a joint representation, establishing a connection between

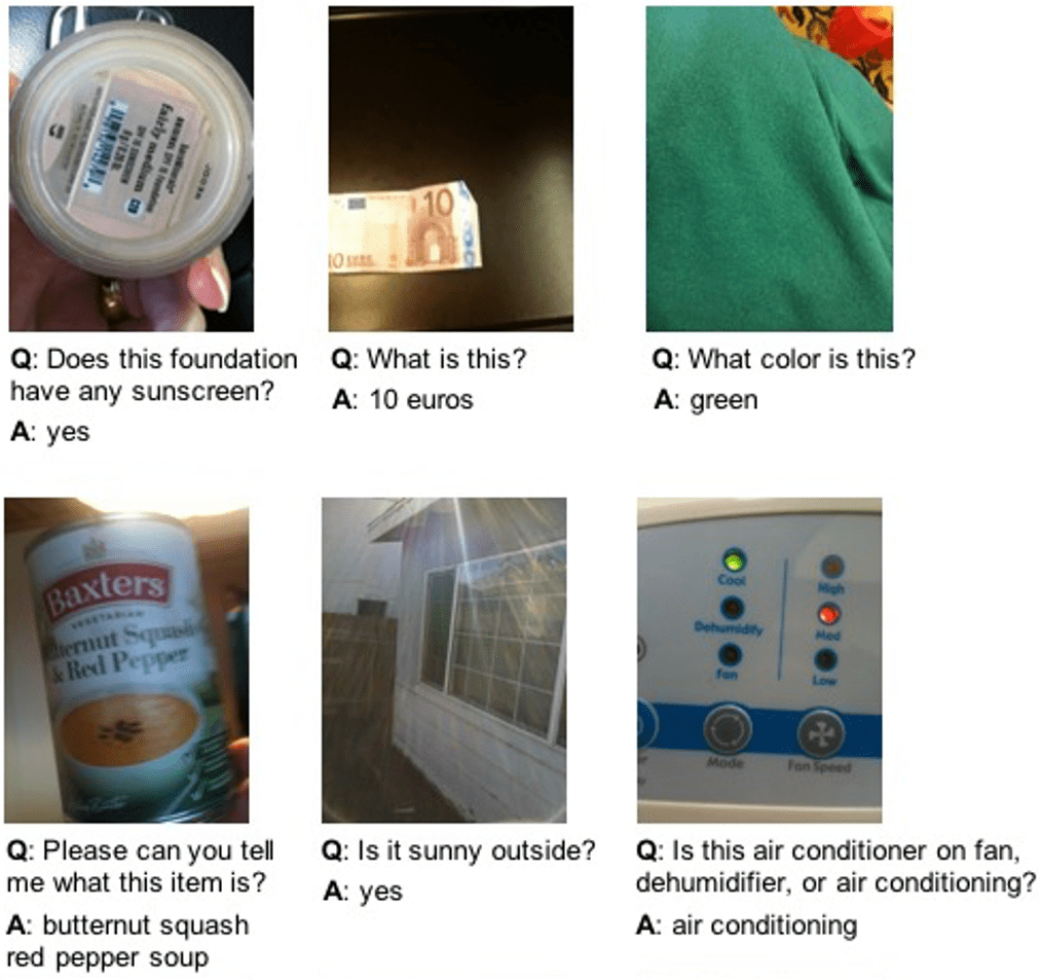


Figure 1.1: Sample from VizWiz Dataset: Image, Question, and Answer from Visually Impaired Users [1].

the image content and the posed question. This joint representation serves as the basis for answer prediction through a mechanism specific to the model employed. Training the system involves datasets containing image-question-answer triplets, allowing the model to learn associations between visual content and textual queries, enabling accurate responses across various contexts [8].

1.2.3 Brief history of VQA System

VQA has experienced significant evolution since its inception. In the 1970s, early work laid the foundation, exploring image understanding and natural language processing as distinct domains. The 1990s saw the concept of a Visual Turing Test, proposed by Turing

in 1950, gaining traction, advocating for machines to comprehend and respond to visual information akin to humans. Advancements in basic image understanding occurred in the 2000s with datasets like PASCAL VOC and the introduction of Scale-Invariant Feature Transform (SIFT) features. The transformative period for VQA unfolded in the 2010s, marked by the launch of the VQA challenge in 2012, serving as a standardized benchmark. The advent of deep learning between 2013 and 2015 revolutionized both image understanding and natural language processing, propelling VQA forward. The year 2016 saw a boost in performance with the introduction of attention mechanisms, particularly in Transformers. Significant moments during the decade included Google's DeepQA system excelling in 2015 and Facebook AI's Mask Region-based Convolutional Neural Network (Mask-R-CNN) showcasing prowess in object detection and captioning in 2016.
The VQA 2.0 challenge in 2022 introduced more diverse and challenging questions, and ongoing research explores aspects such as explainability, robustness, and integration with modalities like touch and language. This narrative encapsulates the pivotal milestones shaping the trajectory of Visual Question Answering throughout its dynamic history.

1.2.4 Motivation

The motivation of VQA lies in our shared human desire to bridge the gap between machines and our natural ways of understanding the world. It's like teaching computers to see and comprehend images just as effortlessly as we do. VQA envisions a future where technology becomes an intuitive companion, responding to our questions about visual content with the ease of a conversation. Beyond this, it's about making technology inclusive — assisting those who are visually impaired by providing them with a way to explore and understand the visual world. Also, VQA is like a playground for researchers, pushing the boundaries of what artificial intelligence can achieve, fostering innovation that can eventually touch every aspect of our lives. So, in a nutshell, VQA is all about infusing a touch of humanity into our machines, making them not just smart but relatable and helpful in our day-to-day interactions [8, 10, 11].

1.2.5 Real-world Applications

The potential application of VQA is constantly expanding as the technology evolves. VQA research promise us to make our lives easier, safer, and more informative by bridging

the gap between human and machine understanding of the visual world. These are just a few examples.

1.2.5.1 Accessibility

- Assistance for visually impaired individuals: VQA-powered applications can describe scenes, identify objects, and answer questions about the physical world, enhancing independence and mobility [12].
- VQA applications can also help image understanding by reading descriptions of images on social media, websites, or documents, providing access to visual information [12].

1.2.5.2 Education

- Interactive museum exhibits: Visitors can ask questions directly about paintings or artifacts, receiving insightful answers generated by AI [13].
- Personalized learning experiences: Students can ask questions about images in textbooks or educational videos, getting instant clarifications or deeper understanding [13].
- Accessibility for visually impaired students: VQA systems can describe images and answer questions, aiding visually impaired students in their learning journey [13].

1.2.5.3 Healthcare

- Medical image analysis: VQA models can assist doctors in analyzing medical images like X-rays or MRIs, identifying potential abnormalities and highlighting areas of concern [14].
- Patient education: Patients can ask questions about their medical images, receiving understandable explanations generated by AI in collaboration with healthcare professionals [14].

1.2.5.4 Robotics

- Scene understanding for robots: VQA-powered robots can better understand their surroundings, identify objects, and make informed decisions based on visual information [15].
- Safe navigation for autonomous vehicles: Vehicles can "ask questions" about their environment, like "Is that a pedestrian crossing the road?" or "What is the speed limit on this sign?", leading to safer navigation.

1.2.5.5 Entertainment

- Interactive games and experiences: VQA can be incorporated into games where players ask questions about virtual environments or images, creating more engaging and dynamic experiences [16].
- Image-based social media interactions: Platforms can use VQA to generate captions or answer questions about user-uploaded images, enriching social media engagement [16].

1.2.5.6 E-commerce

- Personalized product recommendations: VQA systems can analyze images of clothing or home decor and suggest similar items based on user preferences and questions [17].
- Virtual shopping assistants: Customers can ask questions about product details, materials, or styling directly through images, improving the shopping experience [17].

1.2.5.7 Military

- VQA systems could analyze aerial imagery or video feeds from drones and other sources, identifying objects, people, and activities of interest in real-time. This could help commanders make better decisions and improve battlefield awareness.
- VQA could be used to automate tasks like inspecting equipment for damage or identifying spare parts needed for repairs, improving efficiency and reducing downtime.

1.3 Datasets

Several datasets have been suggested for VQA research, each comprising triples consisting of an image, a question, and its corresponding correct answer at the minimum. Some datasets include additional annotations like image captions, regions in the image supporting the answers, or multiple-choice candidate answers. The complexity of datasets and the questions they contain varies, encompassing differences in reasoning requirements and the need for nonvisual information, such as "common sense," to deduce the correct answers. This section offers a thorough comparison of these datasets, discussing their appropriateness for evaluating different aspects of VQA systems. The focus in this section is exclusively on general classical VQA datasets, with other specialized VQA datasets for domains like Medical VQA, TextVQA, and knowledge-based VQA covered elsewhere. Below are examples of some widely used datasets in VQA [8].

1.3.1 Microsoft COCO

The MS COCO (Microsoft Common Objects in Context) dataset is a large-scale resource for object detection, segmentation, key-point detection, and captioning, featuring 328,000 images. Originally released in 2014 with 164,000 images, it underwent changes in 2017, adjusting the training/validation split to 118,000/5,000. The dataset includes annotations for object detection, captioning, keypoints detection, stuff image segmentation, panoptic segmentation, and dense pose annotations. The latter provides detailed information about body poses and shapes for over 39,000 images and 56,000 person instances, available for training and validation images only [18].

1.3.2 VQA

VQA-v1, widely used in VQA research, is based on the COCO dataset, consisting of VQA-v1-real with natural images and VQA-v1-abstract with synthetic cartoon images. VQA-v1-real has 123,287 training images, 81,434 testing images, and diverse question/answer pairs. It includes 614,163 binary questions with 10 answers each, collected by annotators. However, it has a bias where some questions can be answered without seeing the image. VQA-v1-abstract, designed for higher-level reasoning, features 50,000 clipart scenes and 150,000 questions answered by 10 annotators, using a process similar to VQA-v1-real [12].

1.3.3 VQA v2.0

VQA-v2 is an expanded version of VQA-v1-real, aiming to address bias issues. It balances the dataset by collecting pairs of similar images with different answers for each question. With 204,721 images and 1,105,904 questions (10 answers per question), it doubles the image-question pairs compared to VQA-v1-real. This balanced approach helps reduce biases, discouraging VQA models from relying solely on language patterns for higher scores and encouraging models that prioritize visual understanding and interpretability [8].

1.3.4 Visual Genome

Visual Genome includes VQA data in a multiple-choice format, utilizing 101,174 images sourced from MSCOCO and incorporating 1.7 million question-answer pairs. On average, each image has 17 questions. In contrast to the Visual Question Answering dataset, Visual Genome achieves a more balanced distribution across six question types: What, Where, When, Who, Why, and How. Additionally, the Visual Genome dataset provides 108,000 images with comprehensive annotations for objects, attributes, and relationships [19].

1.3.5 CLEVR

CLEVR is a synthetic Visual Question Answering dataset featuring 3D-rendered object images. It includes 70,000 training images with 700,000 questions, 15,000 validation images with 150,000 questions, and a test set. Questions cover Exist, Count, Compare Integer, Query Attribute, and Compare Attribute tasks. Each scene object is described by four attributes: position, size (large/small), shape (square/cylinder/sphere), material type (rubber/metal), and color (gray/blue/brown/yellow/red/green/purple/cyan), resulting in 96 unique combinations [20].

1.3.6 VizWiz

The VizWiz Visual Question Answering dataset is designed to assist individuals with visual impairments by providing answers to questions about images. It encompasses tasks like TextVQA and VizWiz-Captions. The VQA task involves answering questions about images taken by blind individuals, aiming to enhance accessibility through AI tech-

nologies. In the updated version as of January 10, 2023, terminology in answers has been refined, and the new, larger version from January 1, 2020, includes 20,523 training image/question pairs, 205,230 training answer/answer confidence pairs, 4,319 validation image/question pairs, 43,190 validation answer/answer confidence pairs, and 8,000 test image/question pairs [1].This dataset will be utilized in our work to explore and develop advanced techniques in visual question answering.

Figure 1.2 represents a timeline of most popular VQA datasets found in litterature.

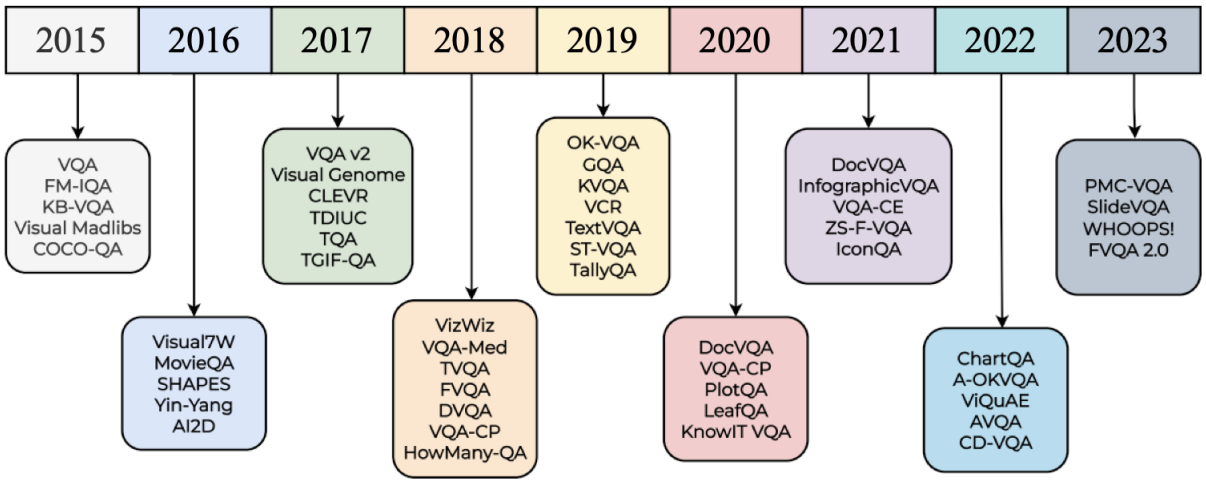


Figure 1.2: Timeline of popular VQA datasets

- The table 1.1 presents an overview of VQA datasets, offering statistics such as the number of images, questions, and the average questions per image.

1.4 Methods and Techniques in VQA

VQA has undergone a transformative journey in terms of methods and techniques, moving from early attempts that relied on handcrafted features to contemporary approaches grounded in deep learning and advanced methodologies. This evolution can be traced across different paradigms, each contributing to the improvement of VQA systems [17, 9, 21].

Table 1.1: Overview of VQA Datasets

Dataset	**Number of images**	**Number of Questions**	**Average of Questions per image**
Microsoft COCO	328,000	-	-
VQA v1.0	204,721	114,163	3
VQA v2.0	265,016	1,457,587	5.4
Visual Genome	101,174	1.7 million	17
CLEVR	70,000	700,000	10
VizWiz	20,528	205,230	10

1.4.1 Early Approaches and Fundamental Techniques

The standard approach involved three key phases: feature extraction, feature conjugation, and answer generation. Feature extraction aimed at distilling meaningful information from multi-modal inputs, relying on traditional methods such as explicit Red-Green-Blue (RGB) vectors, Support Vector Machines (SVM), Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Singular Value Decomposition (SVD) [9].

1.4.1.1 Visual Feature Extraction

For visual features, classical computer vision algorithms like Haar-like features and HOG were common, alongside early CNNs such as LeNet and AlexNet. Transfer learning, particularly with pre-trained CNNs on ImageNet, played a crucial role in adapting these models for VQA tasks [9].

1.4.1.2 Textual Feature Extraction

On the textual side, initial strategies included one-hot vectors and count-based methods. Neural network architectures like Word2Vec, Continuous Bag of Words (CBOW), and Skip-Gram emerged for learning word representations. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) variants, gained popularity for processing questions [9].

1.4.1.3 Feature Conjugation

Feature conjugation involved fusing visual and textual features. Vector operations like Vector Concatenation, Element-Wise Multiplication, and Element-Wise Addition were common choices. Bilinear pooling, as seen in methods like "Multimodal Compact Bi-Linear Pooling" (MCB), became popular for multimodal fusion [9].

1.4.2 Advanced Techniques in VQA

The modern era of VQA is characterized by a shift towards advanced techniques, marking a departure from predefined features to end-to-end learning. Several pivotal methods have emerged, here some popular advanced techniques [9].

1.4.2.1 Attention Mechanisms

Attention mechanisms play a crucial role in enhancing the interpretability and accuracy of VQA models. Inspired by human visual attention, attention mechanisms allow models to focus on specific regions of an image or words in a question, improving the overall reasoning process. Attention mechanisms have become a staple in contemporary VQA architectures, enabling the model to dynamically allocate importance to different parts of the input [9].

1.4.2.2 Transformer-Based Architectures

The introduction of transformers has revolutionized natural language processing and, consequently, VQA. Transformer architectures, initially designed for sequence-to-sequence tasks, have been adapted to handle the fusion of visual and textual modalities. Vision Transformer (ViT) and Cross-Modal Transformers (CMT) are notable examples. Transformers facilitate parallelization of processing, enabling the capture of long-range dependencies and improving the efficiency of VQA models [9].

1.4.2.3 Vision-Language Pre-training (VLP)

Pre-training models on large-scale datasets with diverse tasks, including image captioning and VQA, has become a prevalent strategy. Vision-Language Pre-training (VLP) involves training a model on multiple vision-language tasks before fine-tuning it on downstream

VQA tasks. This approach has shown remarkable success in improving the generalization and performance of VQA models [9].

1.4.2.4 Knowledge-Based Approaches

Acknowledging the importance of external knowledge, recent developments include approaches that integrate knowledge bases into VQA systems. Techniques like Differentiable Graph Neural Networks (GNN) model visual dialogues as structural graphs, providing a framework for incorporating external information into the reasoning process [17].

1.5 Evaluation Metrics

Evaluating VQA systems is uniquely challenging due to the combination of image understanding and natural language processing. Here are some of the key techniques used.

1.5.1 Accuracy based metrics

1.5.1.1 Accuracy

This fundamental metric measures the percentage of correctly predicted answers, providing a straightforward assessment of overall correctness.

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{TrueNegative} + \text{FalsePositive} + \text{FalseNegative}}$$

1.5.1.2 Precision

Precision is an important metric used in VQA evaluation. It helps understand how relevant the model's answers are, and tells the percentage of the predicted answers that are actually correct.

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

1.5.1.3 Recall

Recall tells the percentage of correct answers the model actually predicts. A high recall value indicates that the model is good at finding most of the correct answers.

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

1.5.1.4 F1 Score

Balancing precision and recall, the F1 score is useful for tasks where false positives and false negatives need consideration.

$$\text{F1} = \frac{2\,(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

1.5.2 Semantic similarity metrics

- BLEU (Bilingual Evaluation Understudy): Originally designed for machine translation, BLEU assesses the similarity between predicted and reference answers using n-gram precision.
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Primarily used for text summarization, ROUGE evaluates the overlap in n-grams between predicted and reference answers, emphasizing recall.
- CIDEr (Consensus-based Image Description Evaluation): Adapted for VQA, CIDEr considers consensus among human annotators in evaluating the quality of generated answers.

1.6 Conclusion

In conclusion, this chapter offers a comprehensive exploration of this interdisciplinary field. It begins by introducing the importance of VQA in the convergence of computer vision and natural language processing, providing clear definitions and outlining real-world applications and goals.

Datasets, ranging from COCO to specialized ones like VizWiz, are discussed as essential foundations for VQA research. The methods and techniques section delves into the innovative approaches employed in VQA models, including convolutional and recurrent neural networks, attention mechanisms, and multimodal transformers models.Evaluation metrics such as accuracy, BLEU and CIDEr are highlighted, emphasizing the multifaceted nature of assessing VQA models.

In the next chapter, we will explore the Transformers Architecture, a cornerstone in deep learning for text and image processing.

Chapter 2

Transformers Architecture

2.1 Introduction

Deep learning, particularly with the advent of Transformers, has sparked a paradigm shift in artificial intelligence, notably in natural language processing (NLP) and computer vision (CV). This chapter explores the transformative architecture of Transformers, which has redefined how sequential data is processed. Before delving into Transformers, it's essential to understand the broader context of deep learning's applications in text and image processing like RNN and CNN.

This chapter focuses on the main transformer architecture, the attention mechanism. This mechanism serves as the linchpin of Transformer architectures, enabling models to capture intricate dependencies and contextual nuances. As we navigate through Transformer-based models for NLP and CV, we dissect the components and mechanisms that fuel their remarkable capabilities. From attention mechanisms to multi-head architectures, we delve into how Transformers like Vision Transformer (ViT) and Vision-Language Transformer (ViLT) have expanded the horizons of Deep Learning.

Finally, we will review an overview of related works in the field of VQA aimed at assisting blind and visually impaired individuals, with a focus on research conducted using the VizWiz dataset.

2.2 Deep Learning for Text and Image

Deep Learning techniques have garnered widespread acclaim for their remarkable performance in processing both textual and visual data. In the domain of Natural Language Processing (NLP), models empowered by Deep Learning, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers, have surpassed traditional approaches, exhibiting unprecedented proficiency in tasks such as language translation, sentiment analysis, and text generation. These architectures leverage the power of hierarchical feature extraction, sequential learning, and attention mechanisms to glean insights from text data, enabling unprecedented levels of accuracy and fluency. Similarly, in the realm of Computer Vision (CV), Deep Learning has enabled breakthroughs in image classification, object detection, and image generation. Convolutional Neural Networks (CNNs) have emerged as a cornerstone technology, revolutionizing the way computers perceive and interpret visual information. Meanwhile, Recurrent Neural Networks (RNNs) have proven instrumental in capturing temporal dependencies in sequential data, making them invaluable for tasks like video analysis and captioning. Moreover, Transformers have gained prominence for their ability to efficiently process long-range dependencies in both text and image data, paving the way for novel applications and breakthroughs in AI research. Collectively, these advancements have propelled the field of Deep Learning towards new horizons of possibility, bridging the gap between textual and visual understanding and unlocking unprecedented levels of performance in real world applications [22, 23].

2.2.1 Neural Networks for NLP

In recent years, Natural Language Processing (NLP) has witnessed remarkable progress through the integration of Deep Learning techniques, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. These advancements have fundamentally reshaped how machines understand and generate human language, especially in sequential data [24].

2.2.1.1 The recurrent neural networks (RNN)

RNNs, distinguished by their recurrent connections enabling information persistence over time, have emerged as pivotal tools in NLP tasks such as language modeling. Their ability to capture context and sequential dependencies makes them well-suited for tasks like sentiment analysis and text generation. However, traditional RNNs are challenged by the vanishing gradient problem, hindering their capacity to retain information over long sequences, thus limiting their effectiveness in capturing long-term dependencies [25].

2.2.1.2 Long Short Term Memory (LSTM)

To address the limitations of traditional RNNs, LSTM networks were introduced, featuring a more sophisticated memory cell architecture with input, forget, and output gates. These gates allow LSTMs to selectively retain or forget information over long sequences, thereby facilitating the capture of long-term dependencies in text. In the domain of NLP, LSTM networks have become indispensable, demonstrating superior performance in various tasks such as sentiment analysis, machine translation, and text summarization. Their nuanced understanding of dependencies within text enables coherent responses in conversational AI systems and effective sentiment analysis in social media content. Furthermore, LSTM-based sequence-to-sequence models have facilitated breakthroughs in translation and summarization tasks by encoding and decoding text into a latent representation space. Despite their successes, RNNs and LSTMs face challenges with capturing dependencies over extremely long sequences and slow training times due to their sequential nature. Ongoing research aims to overcome these limitations while further advancing the capabilities of recurrent neural networks in NLP [26].

2.2.2 Neural Networks for CV

Computer Vision (CV) endeavors to imbue computers with the capacity to comprehend and process visual information. Traditionally, this domain relied heavily on meticulously crafted features and algorithms to execute tasks such as object detection and image classification. However, the advent of Deep Learning, particularly Neural Networks, has precipitated a trans-formative shift in CV. These sophisticated algorithms, drawing inspiration from the intricate workings of the human brain, possess the remarkable ability to glean intricate patterns directly from extensive image data. Unlike conventional methods, Neural

Networks avoid explicit feature engineering, instead autonomously uncovering these features through interconnected layers of nodes. This progressive extraction of higher-level information from raw pixels has propelled the efficacy of CV systems to unprecedented heights [27].

2.2.2.1 Convolutional Neural Networks (CNN)

Among the myriad architectures fueling advancements in CV, Convolutional Neural Networks (CNNs) stand out as one of the most successful and impactful. Specifically designed to process grid-like data, CNNs employ convolutional layers adept at efficiently capturing spatial relationships between pixels. By applying filters that slide across the image, convolutional layers identify local features like edges and textures, which are amalgamated to discern more complex features in subsequent layers. This intrinsic capability to learn features automatically, coupled with the robustness afforded by specialized layers like pooling layers for downsampling and retaining essential information, has rendered CNNs indispensable in various CV tasks. Their applications span image classification, object detection, image segmentation, and even image generation, underpinning their pivotal role in reshaping the landscape of computer vision [28].

2.3 Transformers

Transformers are a class of powerful neural network architectures that have revolutionized natural language processing (NLP) tasks by focusing on relationships within sequential data, such as text. Unlike traditional models like recurrent neural networks (RNNs), transformers employ self-attention mechanisms to efficiently identify relevant parts of a sequence for a given task, leading to faster training and better performance. This model architecture consists of a multi-head self-attention mechanism combined with an encoder-decoder structure. Originally designed for NLP, transformers have expanded their impact to computer vision tasks, with models like Vision Transformers (ViTs) adapting the architecture to analyze relationships within images. Additionally, the emergence of multimodal transformers combines the strengths of NLP and computer vision, enabling models to understand and analyze both text and images simultaneously, opening up new possibilities for AI applications. Despite computational challenges, transformers offer a

more nuanced understanding of language and visual data, driving advancements in AI across various domains [2, 3].

2.3.1 Attention Mechanism

Attention mechanisms are pivotal components enhancing model performance, notably in tasks involving sequential data like natural language processing and computer vision. These mechanisms allow models to selectively focus on pertinent parts of input data while disregarding irrelevant information, mimicking human prioritization. By allocating varying weights to different input elements, attention mechanisms dynamically adjust during computation, enabling the model to attend to different parts of the input sequence with variable emphasis. Self-attention, a common type, facilitates capturing long-range dependencies and relationships within the same sequence, while cross-attention extends this capability to different sequences or modalities [2].

2.4 Transformer-based models for NLP

Most competitive neural sequence transduction models utilize an encoder-decoder framework. In this setup, the encoder converts an input sequence of symbol representations $(x_1, \ldots, x_n)$ into a sequence of continuous representations denoted as $z = (z_1, \ldots, z_n)$. Subsequently, the decoder generates an output sequence $(y_1, \ldots, y_m)$ of symbols iteratively, where each symbol is produced one at a time. Throughout this process, the model operates in an auto-regressive manner, incorporating previously generated symbols as additional input when predicting the next symbol. The Transformer model adheres to this general architecture, employing stacked self-attention and point-wise, fully connected layers for both the encoder and decoder components [2], as illustrated in Figure 2.1.

2.4.1 Word Embeddings

The first step in Figure 2.1 is word embedding. This process is crucial for converting input sequences into machine-readable representations, efficiently capturing contextual information. Unlike one-hot encoding, which results in large, sparse vectors with minimal information, word embeddings transform these vectors into dense, lower-dimensional representations, considering contextual nuances, the original paper [2] use 512 dimensions

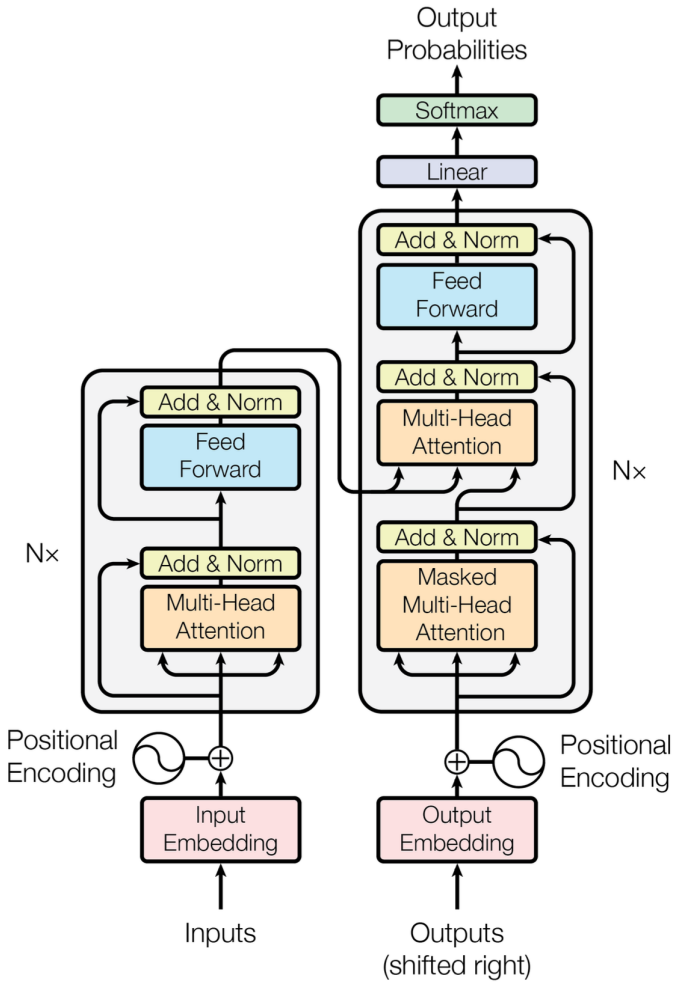


Figure 2.1: Transformer model architecture [2].

of embedding vector (d_model) [2]. This transformation enhances the model's ability to understand the semantic relationships between words within a given context. In the Transformer architecture, word embedding serves as the initial component, reducing dimensionality while preserving contextual dependencies between words. It acts as a lookup table, mapping input vectors to a lower-dimensional space and enabling the model to capture relationships and contextual nuances between words [2].

2.4.2 Positional Encoding

The Transformer architecture, which replaces recurrence-based networks with self-attention mechanisms for processing input sequences, offers faster training and improved handling of long-range dependencies but lacks inherent information about the relative positions of

words. To remedy this, positional encoding is introduced, augmenting each word embedding vector with a unique vector of length *d_model* determined by the word's position in the input sequence. This encoding enables the model to discern the relative positions of words and integrate this spatial information into its processing. Specifically, the positional encoding formula 2.1 incorporates parameters "pos" representing word position and "i" indicating the position of values within the word embedding. By applying both positional encoding functions (formula 2.1 for even positions and formula 2.2 for odd positions), two distinct values are generated for each "i" value, ensuring comprehensive coverage across the embedding dimension (d_model). For instance, in a sentence like "What is this?" with a d_model of 512, the word "this" would have a pos value of 3, while its "i" value ranges from 0 to 255 [2].

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right) \tag{2.1}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right) \tag{2.2}$$

2.4.3 Encoder

The encoder shown in the left half of Figure 2.1 in the Transformer architecture is responsible for converting input sequences into machine-readable representations that capture both word similarity and relative positional information. Comprised of a stack of identical layers, typically six, each layer contains two sub-layers: a multi-head self-attention mechanism and a positionwise fully connected feed-forward network. Residual connections and layer normalization are applied around each sub-layer to facilitate information flow and mitigate potential gradient issues. Leveraging multi-head attention mechanisms, the encoder discerns intricate word relationships, akin to human language analysis. Finally, the accumulated knowledge from the encoder is passed to the decoder for generating the final output sequence [2].

2.4.4 Decoder

The transformer decoder architecture is utilized in tasks such as language generation, where the model is tasked with producing a sequence of words given an input prompt or context. Unlike the encoder, the decoder operates in a step-by-step manner, generating

each word conditioned on the previously generated words. To facilitate this process, the decoder employs a technique known as triangle masking for attention. This technique restricts the attention mechanism to only consider tokens to the left of the current token being generated, preventing the model from accessing tokens it hasn't yet produced and ensuring a more coherent generation process [2].

2.4.5 Attention

An attention function can be defined as a process that maps a query and a collection of key-value pairs to produce an output, with all elements—query, keys, values, and output—represented as vectors. The output is determined by computing a weighted sum of the values, where each value's weight is determined by a compatibility function between the query and its corresponding key [2].

2.4.6 Scaled Dot-Product Attention

Scaled Dot-Product Attention is an attention mechanism designed to operate on queries (Q), keys (K), and values (V), of dimensionality d_k. It involves computing the dot products between the query and all keys, scaling the result down by $\sqrt{d_k}$, and applying a softmax function to obtain weights for the values. In practice, the attention calculation proceeds as follows: Given a query Q, a key K, and a value V, the attention is computed by taking the dot product of Q and K, dividing the result by $\sqrt{d_k}$, and then applying the softmax function to obtain the weights for the values, as illustrated in Figure 2.2, and we calculate the attention as follows:

$$Attention\,(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3}$$

The two main attention mechanisms are additive and dot-product attention. Dot-product attention, scales the compatibility function by $\frac{1}{\sqrt{d_k}}$, while additive attention employs a feed-forward network with a single hidden layer. Despite comparable theoretical complexities, dot-product attention is faster and more space-efficient due to optimized matrix multiplication. While both mechanisms perform similarly for small d_k values, additive attention outperforms dot-product attention without scaling for larger d_k values. This

difference is attributed to potential gradient issues caused by large dot product magnitudes, mitigated by scaling in dot-product attention [2].

2.4.7 Multi-Head Attention

Instead of using a single attention function with $d_{model} - dimensional$ keys, values, and queries, it's advantageous to project queries, keys, and values multiple times with distinct learned linear projections to dimensions d_k, d_k and d_v, respectively. Then, attention functions are applied in parallel on these projected versions, resulting in $d_v - dimensional$ output values. These outputs are concatenated and projected once more to obtain the final values, as illustrated in Figure 2.2. This approach, known as multi-head attention, allows the model to attend to information from different representation subspaces simultaneously at different positions, improving its ability to capture diverse patterns and dependencies. In contrast, using a single attention head and averaging restricts this capability [2], and is defined as:

$$MultiHead\,(Q, K, V) = (head_1, \ldots, head)\,W_0 \tag{2.4}$$

$$\text{where} \quad head_i = Attention\left(QW_i^Q, KW_i^k, VW_i^\nu\right) \tag{2.5}$$

2.5 Transformer-based models for CV

Transformer-based models are revolutionizing Computer Vision (CV) by offering a fresh perspective on processing visual data. Unlike traditional Convolutional Neural Networks (CNNs), transformers excel at capturing long-range dependencies and contextual relationships within images, making them particularly adept at tasks like object detection and image classification. By leveraging self-attention mechanisms similar to those in NLP transformers, these models achieve a more holistic understanding of image content. Although challenges such as computational cost and integration into existing workflows remain, the promise of transformers in CV is undeniable. The Vision Transformer (ViT) model was introduced in 2020 by Google Research, Brain Team in the paper titled "AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE" [3].

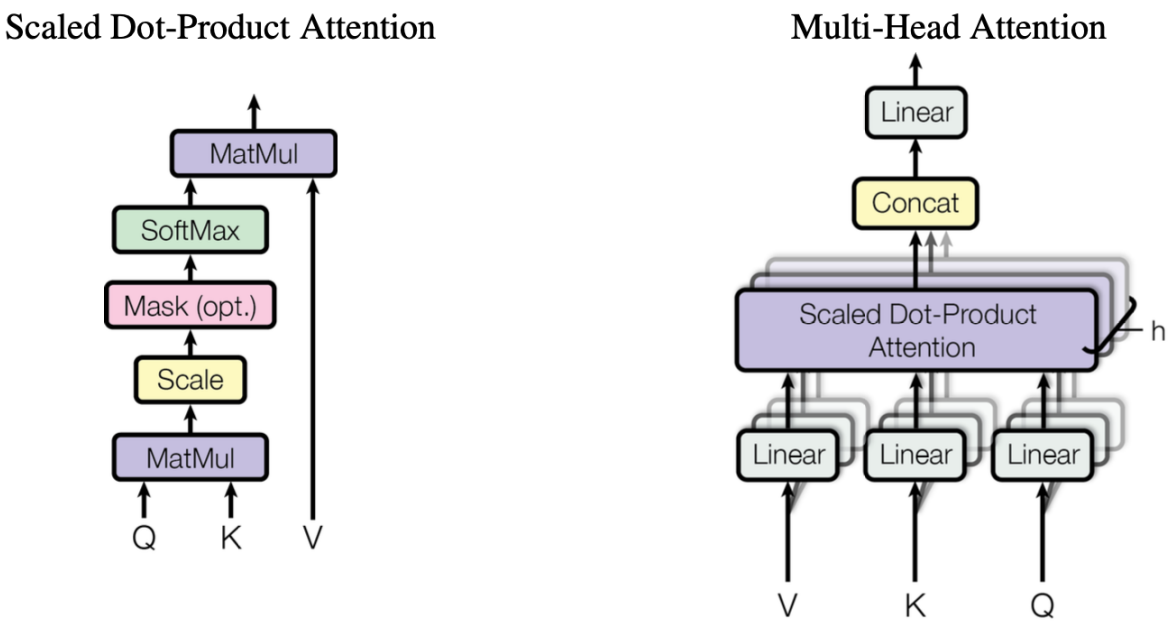


Figure 2.2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel [2].

2.5.1 Vision Transformer Architecture (ViT)

Inspired by the success of Transformer models in natural language processing (NLP), researchers have extended their application to image processing through the Vision Transformer (ViT) architecture. ViT [3] revolutionizes image analysis by treating images as sequences of patches, akin to tokens in NLP. By directly applying the Transformer to image patches, ViT enables holistic understanding and feature extraction [3], as illustrated in Figure 2.3

2.5.1.1 Patch Splitting

The initial step in the Vision Transformer (ViT) architecture involves partitioning images into fixed-sized patches, effectively converting the image into a sequence of manageable segments. This process enables the model to process images in a structured manner, facilitating subsequent analysis [3].

2.5.1.2 Flattening of Image Patches

Once the images are segmented into patches, these patches are flattened to create a more compact and uniform representation. Flattening the patches simplifies the data format,

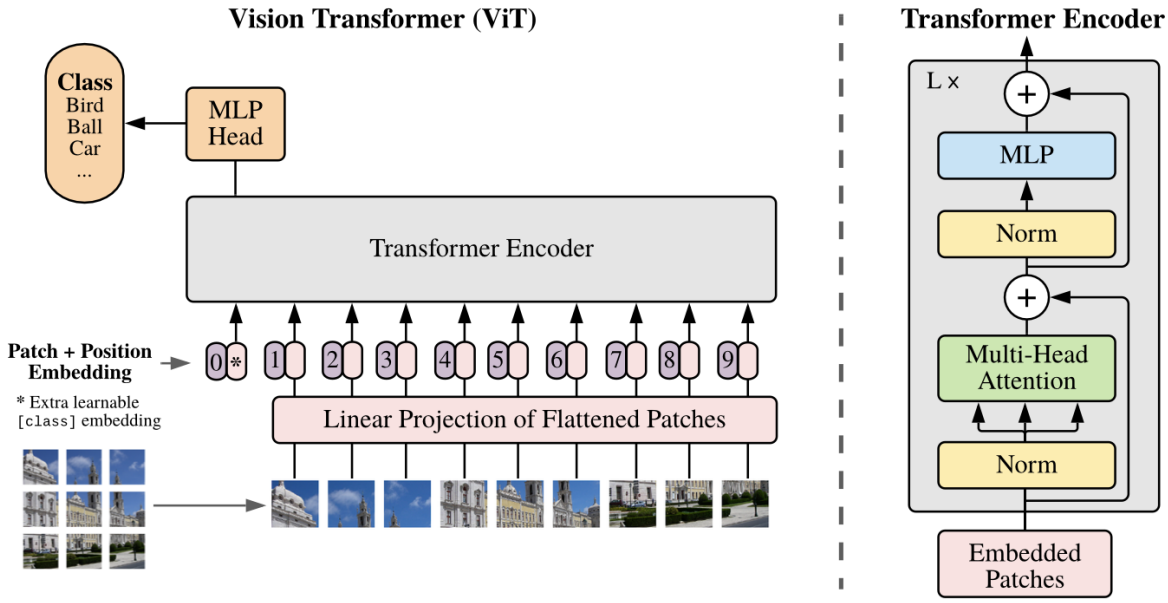


Figure 2.3: Vision Transformer model architecture [3]

making it conducive to processing by subsequent layers in the model [3].

2.5.1.3 Linear Embedding Generation

Following flattening, linear embeddings are generated from the flattened image patches. These embeddings encode essential features of the image patches while reducing computational complexity, ensuring efficient processing throughout the network [3].

2.5.1.4 Inclusion of Positional Embeddings

To incorporate crucial positional information into the model, positional embeddings are introduced. These embeddings denote the spatial arrangement of patches within the image, allowing the model to understand the relative positions of different elements in the image sequence [3].

2.5.1.5 Input to Transformer Encoder

The sequence of patch embeddings, along with positional embeddings, serves as the input to a transformer encoder. This encoder employs self-attention mechanisms to capture complex relationships between patches and refine the representation of the image, facilitating effective feature extraction and information processing [3].

2.5.1.6 Pre-training with Image Labels

The ViT model is pre-trained using a large dataset of labeled images in a fully supervised manner. During this phase, the model learns to extract meaningful features from images and associate them with corresponding labels, enriching its understanding of various visual concepts [3].

2.5.1.7 Fine-tuning for image classification

Finally, the pre-trained ViT model undergoes fine-tuning on downstream datasets tailored for specific tasks, such as image classification. This process allows the model to adapt its learned features to new domains and optimize performance for the task at hand. Through these systematic steps, the ViT architecture revolutionizes image processing by leveraging transformer-based models to achieve exceptional results in diverse computer vision tasks [3].

2.6 Multimodal Transformers Vision Language Models (VLMs)

A vision-language model combines vision and natural language processing capabilities by processing both images and their corresponding textual descriptions. Through this fusion, the model learns to associate information from both modalities, leveraging spatial features from images and encoding textual information. By mapping data from both sources, including detected objects, image spatial layout, and text embeddings, the model gains an understanding of images and can express this knowledge in natural language.In our work, we will utilize a Vision-and-Language Transformer (ViLT) to address tasks such as VQA, where the model takes both images and corresponding questions as inputs. This VLM ingests images alongside textual descriptions, learning to comprehend images and encode this knowledge into natural language.

2.6.1 Vision-and-Language Transformer (ViLT)

The ViLT model, introduced by Wonjae Kim, Bokyung Son and Ildoo Kim in the paper titled "Vision-and-Language Transformer Without Convolution or Region Supervision" [4], is a deep learning architecture that integrates computer vision and natural language processing (NLP) to tackle tasks like VQA and image captioning. It inherits its foundation from the Vision Transformer (ViT) architecture, known for its effectiveness in computer vision tasks as illustrated in Figure 2.4.

ViLT extends the ViT framework by incorporating text embeddings, enabling it to engage in vision-and-language pre-training (VLP). This capability allows the model to glean insights from extensive datasets containing both images and corresponding textual descriptions. ViLT undergoes fine-tuning on the VQAv2 dataset, a widely used benchmark dataset for VQA tasks.

Characterized by its minimal design, ViLT boasts computational efficiency and a relatively modest parameter count compared to other VLP models. These attributes render it suitable for real-world applications where computational resources may be constrained.

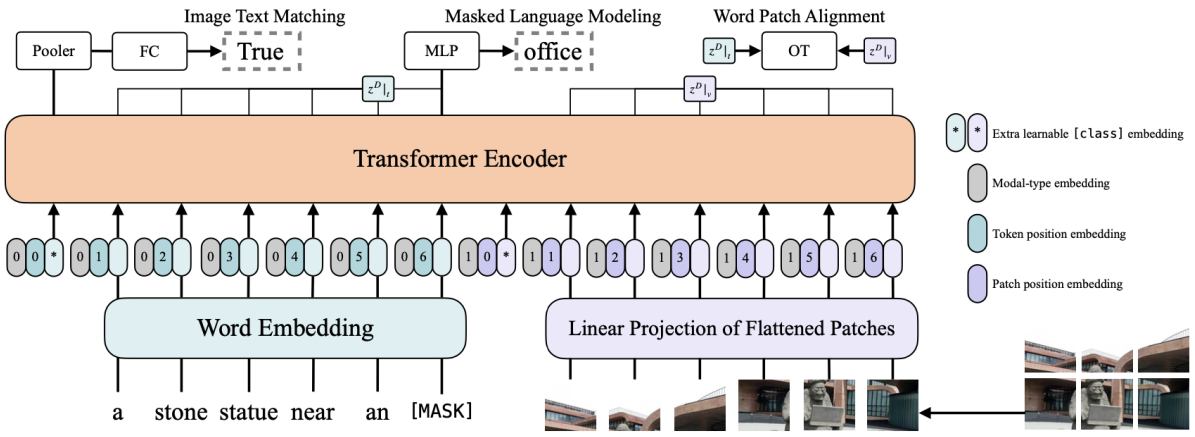


Figure 2.4: Vision Transformer model architecture [4].

2.7 Related Works

This section presents an overview of related works in the field of VQA aimed at assisting blind and visually impaired individuals, focusing on research conducted using the VizWiz dataset. The VizWiz dataset, specifically designed for this purpose, comprises images captured by blind users along with accompanying textual questions. The dataset is unique in that it captures real-world scenarios encountered by blind individuals, providing a diverse and challenging set of images and questions. The VQA challenge associated with the VizWiz dataset consists of two primary tasks: predict the answer to a visual question and predict whether a visual question cannot be answered [29].

Alayrac et al. introduced Flamingo, a Visual Language Model for Few-Shot Learning [30], which encompasses a family of Visual Language Models capable of performing various multimodal tasks, including VQA. Flamingo utilizes pre-trained vision and language models. A key component, the Perceiver Resampler, bridges these models, enabling them to collaborate effectively for tasks such as analyzing images and text to generate comprehensive textual responses. This approach is reflected in its VQA performance results on the VizWiz-QA benchmark in the task of predicting the right answer. It achieves an accuracy of 65.4% when finetuning the largest model.

In [31], Singh et al.proposed the LoRRA (Look, Read, Reason, Answer) model architecture for visual question answering. This architecture consists of three key components: (i) a VQA component, (ii) a reading component, and (iii) an answering module. The VQA component, Pythia v0.3, is a refined implementation of Pythia v0.1 [32], which was the winner of the VQA 2018 challenge. Pythia v0.3 extracts image features, both grid-based and region-based, and processes question text using pre-trained word embeddings and an LSTM with self-attention. Notably, it integrates an explicit reading component by incorporating pre-trained FastText embeddings for Optical Character Recognition (OCR) outputs. Finally, an answer module combines these processed information streams to generate the final answer. Evaluation of a single model on the VizWiz dataset yields an accuracy of 54.72%.

Chen et al [33]. introduced and developed PaLI (Pathways Language and Image Model), a large language model (LLM) designed for joint image and text processing, leverages pre-trained components for efficiency. At its core, PaLI combines a large ViT model for image understanding with a pre-trained mT5 model for text processing. PaLI is trained on a massive, multilingual dataset called WebLI, along with other resources, to tackle various image-language tasks. PaLI achieves increasing accuracy on the VizWiz-QA benchmark as model size scales, achieving 67.5% accuracy for the 3B parameter model, 71.1% for the 15B parameter model, and 73.3% for the 17B parameter model.

In their paper, Deuser et al [34]. propose a simplified VQA approach that leverages a pre-trained CLIP model for both image and text encoding. Their approach utilizes a CLIP ensemble consisting of two robust encoders: a high-resolution ResNet-50 scaled by 64x (RN50x64) and a ViT-L/14 model resized to 336x336 pixels. Instead of training a complex model from scratch, they train an additional linear classifier head on top of the pre-trained CLIP. Their strategy incorporates three key components: (i) curating a specific vocabulary of common answers to enhance classification performance, (ii) employing linear layers directly on CLIP's pre-extracted image and text features for VQA tasks, and (iii) introducing an optional element, an answer type gate, that facilitates a learnable masking mechanism to further refine answer prediction. Evaluated on the VizWiz 2022 VQA Challenge, the model achieves an accuracy of 60.15%.

The authors in [35] propose a novel approach that decouples box proposal and featurization for VQA. This allows separate training in region identification (box proposal) and feature extraction (featurization), promoting better transfer learning. They leverage a simplified "up-down" model architecture based on Pythia v0.1. During training, the model utilizes standard Faster R-CNN (B-FRCNN) based bounding box features. However, for evaluation, these are replaced with Ultra-based features (B-Ultra). The reported accuracy on the test-standard split for the VQA task on the VizWiz dataset using B-FRCNN is 51.9 % and the B-Ultra is 53.7%.

2.7.1 Comparative table

The following table 2.1 summarizes the related works discussed previously and compares them to our proposed work based on different comparison criteria.

Work	**Model**	**Training/ Finetuning Dataset**	**Evaluation Dataset**	**Scores [Acc (%)]**	
				test-dev	**test-std**
Alayrac et al [30]. 2022	Flamingo	VizWiz	VizWiz	65.7	65.4
Singh et al [31]. 2019	Pythia v0.3	ImageNet, Visual Genome, GloVe embeddings datasets	VizWiz	/	54.72
Chen et al [33]. 2022	PaLI-17B	VQAv2, OKVQA, TextVQA, VizWiz	VizWiz	**74.4**	**73.3**
Deuser et al [34]. 2022	CLIP	/	VizWiz	61.64	60.15
Soravit et al [35]. 2019	B-FRCNN	/	VizWiz	/	51.9
	B-Ultra				53.7
Proposed	PALIGemma	VizWiz	VizWiz	**80.00**	/

Table 2.1: Comparative table of related works

2.8 Conclusion

In this chapter, we have explored the intersection of deep learning with both text and image processing. We began by discussing the fundamentals of neural networks for natural language processing (NLP) and computer vision (CV). We gained insight into their respective roles and applications in analyzing textual and visual data. Following this, we introduced transformers, a powerful architecture that has transformed various NLP tasks, and delved into their key components, including attention mechanisms, word embeddings, and positional encoding. Moreover, we examined the adaptation of transformer-based models for both NLP and CV tasks. Notable models such as the Vision-and-Language Transformer (ViLT) were explored, showcasing the potential of integrating vision and language processing in a unified framework.

Finally, we reviewed the related works in the field of VQA aimed at assisting blind and visually impaired individuals, focusing on research conducted using the VizWiz dataset. With this foundational knowledge, we are now prepared to delve into our proposed approach in the next chapter.

Chapter 3

Experimental and Proposed Solution

3.1 Introduction

This chapter embarks on a detailed comparative study of four state-of-the-art multimodal VQA models: Vision-Language Transformer [4] (ViLT), Contrastive Language-Image Pre-training [5] (CLIP), Bootstrapping Language-Image Pre-training [6] (BLIP), and Google PALIGemma [7, 33]. Our investigation begins with an in-depth data exploration of the VizWiz dataset, which is specifically designed for VQA tasks aimed at assisting blind and visually impaired individuals. This dataset provides a robust foundation for evaluating and benchmarking the performance of our selected models.

The chapter will systematically cover several critical components to offer a comprehensive comparison. First, we delve into the architecture and language models underpinning each VQA system, examining how these models integrate and process visual and textual information. Next, we discuss the preprocessing techniques employed to prepare the data for each model, ensuring that the inputs are optimally formatted for effective learning.

We will then explore the hyperparameters that govern the training process of each model, alongside the fine-tuning strategies used to adapt pre-trained models to the VizWiz dataset. Performance metrics, such as accuracy and other relevant evaluation measures, will be outlined to provide a clear framework for assessing model performance.

Our experimental setup section will detail the methodology and environment in which the models are trained and tested, ensuring reproducibility and transparency in our comparative analysis. We will follow this with a thorough comparison and discussion of the results, highlighting the strengths and weaknesses of each model in the context of the

VizWiz dataset.

Finally, we will test the models against various scenarios and problems inherent in the VQA task, addressing challenges such as ambiguity, contextual understanding, and the ability to generalize across different types of questions and images. Through this comparative study, we aim to provide valuable insights into the current capabilities and limitations of multimodal VQA models, paving the way for future advancements in this dynamic field.

3.2 Used Dataset

To train our models, we use the VizWiz Visual Question Answering dataset [29] as we mentioned in chapter one. This dataset is specifically designed to assist individuals with visual impairments by providing answers to questions about images they have taken.

3.2.1 Dataset Exploration of VizWiz

The VizWiz Visual Question Answering dataset is structured in four main folders: Annotations, Train, Val, and Test. The Annotations folder contains three JSON files—train.json, val.json, and test.json—which are crucial for organizing the data for the training, validation, and testing phases. The Train folder holds the training images, the Val folder holds the validation images, and the Test folder holds the test images.

3.2.1.1 Content of JSON Files

Each JSON file in the Annotations folder has specific content. The train.json file contains 20,523 entries, each with an image filename, a question, a list of answers, and metadata. The training set has 41,229 distinct answers. The val.json file contains 4,319 entries with a similar structure and 10,905 distinct answers. The test.json file contains 8,000 entries that include only the image filenames and questions. Examples of these entries are shown in Figure 3.1. The table 3.1 provides a clear summary of the number of entries and distinct answers in each dataset.

The following figure 3.2 illustrates a concrete example taken from the VizWiz dataset for illustration purposes.

Dataset	Total Samples	Distinct Answers
Train	20,523	41,229
Validation	4,319	10,905
Test	8,000	-

Table 3.1: Summary of the VizWiz dataset entries and distinct answers.

```
VizWizDic({
    Train: Dataset({
        features: df[['image', 'question', 'answers', 'answer_type', 'answerable']],
        nbr_rows: 20523
})
    Validation: Dataset({
        features: df[['image', 'question', 'answers', 'answer_type', 'answerable']],
        nbr_rows: 4319
})
})
```

Figure 3.1: Dataset format


```
{
    "image": "VizWiz_train_00000000.jpg",
    "question": "What's the name of this product?",
    "answers": [
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil"},
      {"answer_confidence": "yes", "answer": "basil"},
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil leaves"},
      {"answer_confidence": "yes", "answer": "basil"}
    ],
    "answer_type": "other",
    "answerable": 1
}
```


Figure 3.2: Input Example from vizwiz

3.2.1.2 Distribution of Answer Types and Answerability

The Train, Val, and Test folders contain the images referenced in their respective JSON files. We visualize the distribution of *answer_type* and *answerable* columns in the training and validation datasets. For the training datasets, the distribution of *answer_type* includes 13,733 "other" answers, 5,532 "unanswerable" answers, 957 "yes/no" answers, and 301 "number" answers. The distribution of answerable indicates 14,991 entries marked as answerable and 5,532 as unanswerable. The training set has 41,229 distinct answers and a total of 20,523 samples.

For the validation dataset, the *answer_type* distribution includes 2,691 "other" answers, 1,385 "unanswerable" answers, 195 "yes/no" answers, and 48 "number" answers. The answerable distribution shows 2,934 answerable entries and 1,385 unanswerable entries. The validation set has 10,905 distinct answers and a total of 4,319 samples. These visualizations help us understand the distribution and frequency of different types of answers and answerability within the dataset, providing a foundation for further analysis and model training.

3.2.1.3 Length Analysis of Questions and Answers in the VizWiz Dataset

The analysis of the VizWiz dataset's questions and answers reveals interesting insights into their length distributions. The questions exhibit a broad range, with an average length of approximately 32 characters. The median length is 25 characters, and most questions (75%) are 36 characters or shorter. However, the longest question reaches an impressive 302 characters, showcasing the dataset's diversity. In contrast, answers are generally more concise, with an average length of about 10 characters. The median length of answers is 10 characters, and 75% of answers are 12 characters or shorter, with the longest answer extending to 93 characters. These statistics highlight that while questions can be quite detailed and lengthy, answers tend to be brief and to the point, reflecting the nature of visual question answering tasks where concise and specific responses are often sufficient, illustrated in figures [3.3, 3.4].

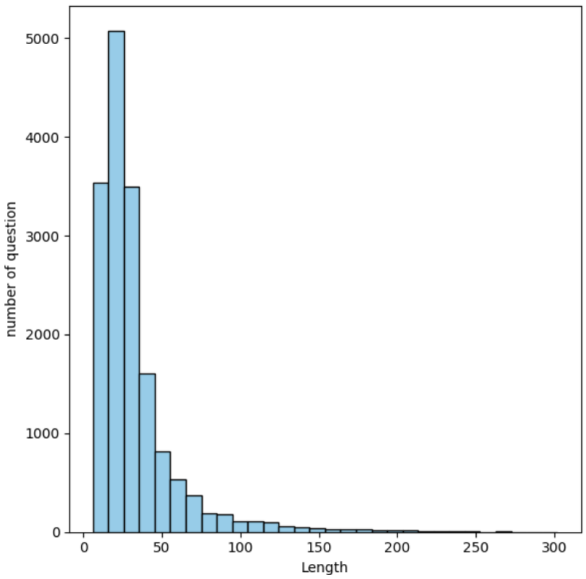


Figure 3.3: Distribution of Question Lengths

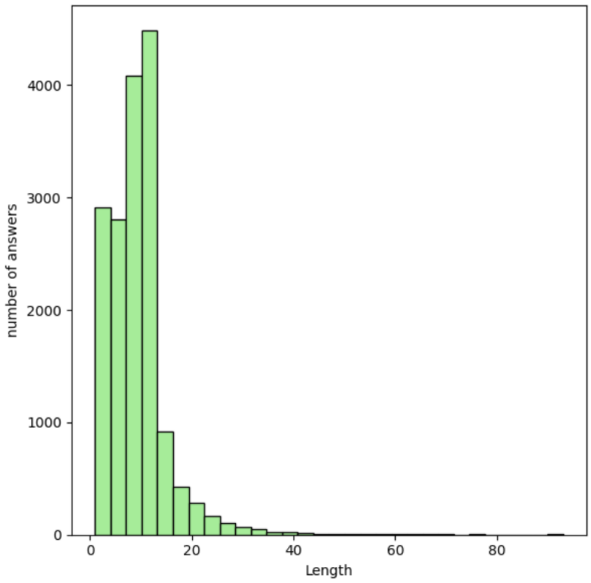


Figure 3.4: Distribution of Answer Lengths

3.2.1.4 Train-Validation Split

For our experiments, we use the Train dataset only by splitting it into 80% for training and 20% for validation. This results in a training set with 13,134 samples and a validation set with 3,284 samples. For the training set, the distribution of answer_type includes 8,789 "other" answers, 3,518 "unanswerable" answers, 622 "yes/no" answers, and 205 "number" answers. The answerable distribution indicates 9,616 entries marked as answerable and 3,518 as unanswerable. The training set has 28,268 distinct answers. For the validation set, the answer_type distribution includes 2,189 "other" answers, 926 "unanswerable" answers, 140 "yes/no" answers, and 29 "number" answers. The answerable distribution shows 2,358 answerable entries and 926 unanswerable entries. The validation set has 8,620 distinct answers.

The figures [3.5, 3.6, 3.7, 3.8, 3.9, 3.10] is statistic comparison between training and validation dataset. The detailed statistics for these splits are presented in the table 3.2.

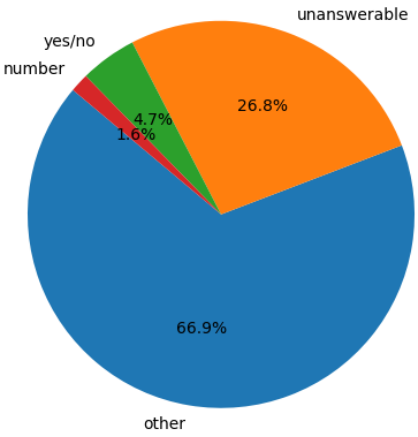


Figure 3.5: Pie chart of training answer type

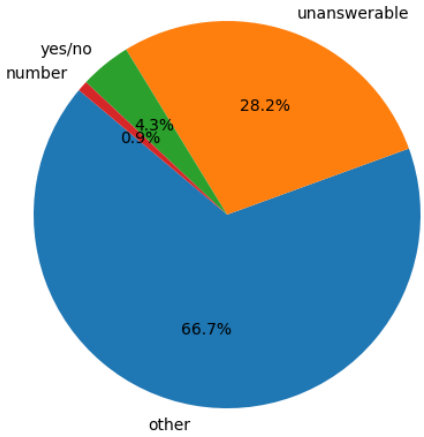


Figure 3.6: Pie chart of validation answer type

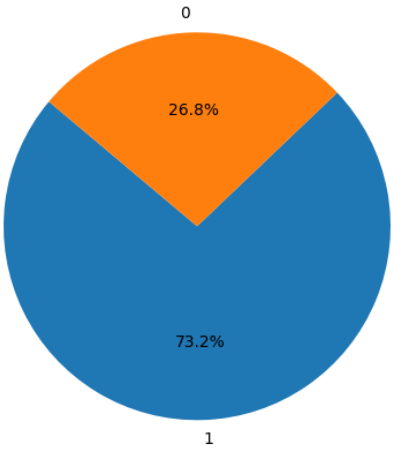


Figure 3.7: Pie chart of training answerable

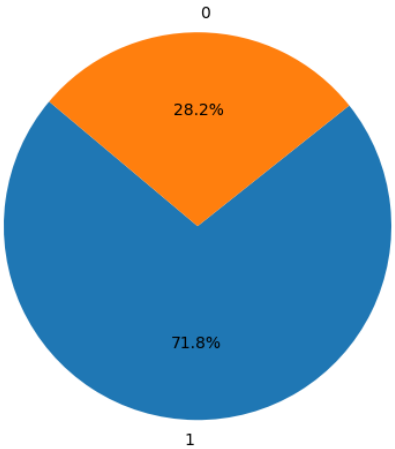


Figure 3.8: Pie chart of validation answerable

3.3 Comparative Study

In this section, we provide a detailed comparative analysis of the four state-of-the-art multimodal VQA models: Vision-Language Transformer (ViLT), Contrastive Language-Image Pre-training (CLIP), Bootstrapping Language-Image Pre-training (BLIP), and Google PaliGemma. This comparative study focuses on various aspects, including the architecture and language models underpinning each VQA system, preprocessing techniques, hyperparameters, fine-tuning strategies, and performance metrics.

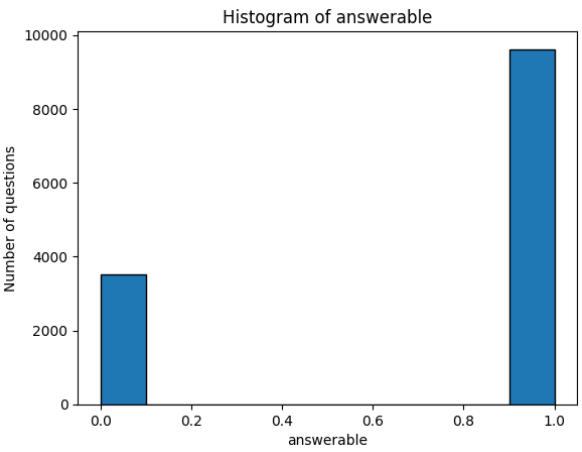


Figure 3.9: Number of answerable questions on training set

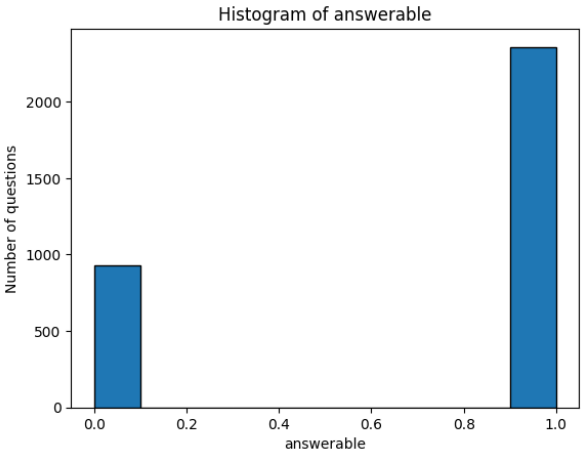


Figure 3.10: Number of answerable questions on validation set

Dataset	Answer Type	Count	Answerable	Count	Total Distinct Answers	Total Samples
Training Set	Other	8,789	Answerable	9,616	28,268	13,134
	Unanswerable	3,518		3,518		
	Yes/No	622				
	Number	205				
Validation Set	Other	2,189	Answerable	2,358	8,620	3,284
	Unanswerable	926		926		
	Yes/No	140				
	Number	29				

Table 3.2: Detailed statistics for the training and validation set.

3.3.1 Language Models

The foundation of any VQA model lies in its ability to effectively process and integrate visual and textual information. This is primarily achieved through sophisticated language models that are jointly trained on both modalities. We will explore the language models used by CLIP, BLIP, and PaliGemma in detail, with a brief reference to ViLT, which was covered in the previous chapter.

3.3.1.1 Vision-Language Transformer (VILT)

As discussed in the previous chapter, ViLT (Vision-Language Transformer) leverages a transformer-based architecture to seamlessly integrate visual and textual inputs. By directly encoding images as sequences of patches, similar to words in text, ViLT avoids the need for convolutional neural networks (CNNs), leading to a more unified and efficient approach to multimodal learning.

3.3.1.2 Contrastive Language-Image Pre-training (CLIP)

CLIP (Contrastive Language-Image Pre-training [5]) is an innovative model designed to predict whether an image and a text snippet from the web are paired, utilizing a large dataset for pre-training. It leverages a multi-modal embedding space by training an image encoder and a text encoder jointly to maximize the cosine similarity between correct image-text pairs while minimizing it for incorrect pairings. Notable for its zero-shot learning capability, CLIP can perform tasks without task-specific training, demonstrating significant performance improvements on various image classification datasets compared to prior models like Visual N-Grams. Its pre-training on diverse web data enables impressive results across different tasks, sometimes even outperforming supervised models. The figure 3.11 illustrates CLIP approach. Unlike standard image models that train an image feature extractor and a linear classifier together to predict a label, CLIP simultaneously trains an image encoder and a text encoder to predict the correct pairings in a batch of (image, text) training examples. During testing, the learned text encoder creates a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

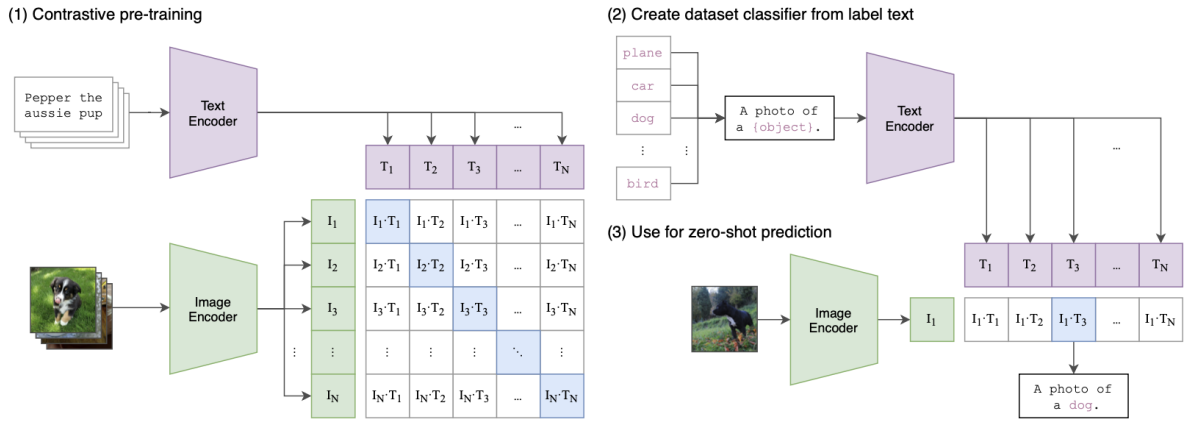


Figure 3.11: CLIP: Efficient Zero-shot Transfer Learning [5].

3.3.1.3 Bootstrapping Language-Image Pre-training (BLIP)

BLIP, which stands for Bootstrapping Language-Image Pre-training [6], is a novel framework designed to enhance both vision-language understanding and generation tasks. Traditional vision-language pre-training models excel either in understanding-based tasks or generation-based tasks, but not both, often relying on large datasets of noisy image-text pairs from the web, which are suboptimal for training. BLIP addresses these challenges by utilizing a multimodal mixture of encoder-decoder architecture, allowing it to function as both a unimodal encoder and an image-grounded text decoder. Furthermore, BLIP incorporates a Captioning and Filtering (CapFilt) mechanism to improve the quality of training data, involving a captioner that generates synthetic captions and a filter that removes noisy ones, enhancing the dataset's quality. The figure 3.12 illustrates the BLIP architecture, integrating several components for effective vision-language tasks, including a text encoder and an image encoder that transform text and image inputs into embeddings. The central Multimodal Mixture of Encoder-Decoder (MED) operates as a unimodal encoder, an image-grounded text encoder, and an image-grounded text decoder. Additionally, the framework includes a Captioner (Cap) and a Filter (Filt), both pre-trained using the MED model and then fine-tuned separately. The architecture supports three vision-language pre-training objectives: image-text contrastive learning (ITC), image-text matching (ITM), and image-conditioned language modeling (LM). BLIP achieves state-of-the-art results across various vision-language tasks, including image-text retrieval, image

captioning, and visual question answering, demonstrating strong generalization capabilities, even in zero-shot transfers to video-language tasks.

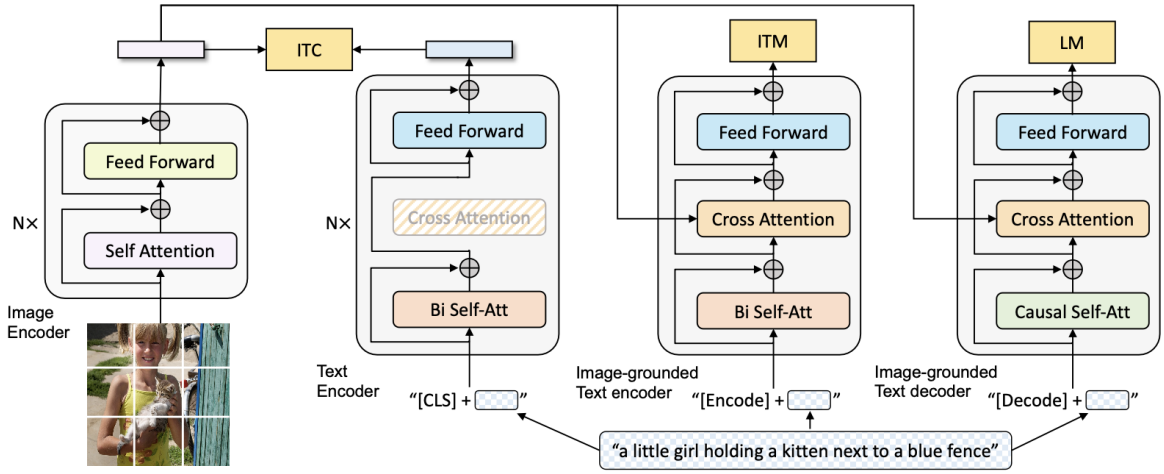


Figure 3.12: Blip model architecture [6].

3.3.1.4 PaliGemma

PaliGemma [7] is a versatile family of vision-language models that combines the state-of-the-art SigLIP-So400m image encoder with the Gemma-2B text decoder, resulting in a powerful tool for tasks involving both images and text. SigLIP functions similarly to CLIP, containing both image and text encoders trained jointly, while Gemma serves as a decoder-only model for text generation. PaliGemma can be fine-tuned for various specific tasks such as image captioning, visual question answering, entity detection, and referring expression segmentation. The models are available in different resolutions and precisions, and come in three types: pretrained (PT), fine-tuned for mixed tasks (Mix), and fine-tuned for specific academic benchmarks (FT). Despite requiring substantial memory for high-resolution inputs, the 224x224 resolution models are generally sufficient for most applications. PaliGemma's capabilities are best leveraged by fine-tuning it to particular use cases, with task prefixes like "detect" or "segment" guiding the model's output. The mix checkpoints demonstrate the model's diverse capabilities and are suitable for interactive testing across various tasks. The following figure 3.13 illustrates PaliGemma system architecture.

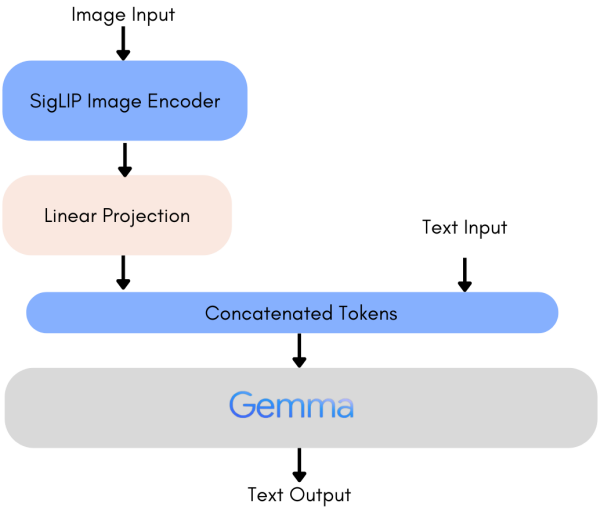


Figure 3.13: PaliGemma model architecture [7].

3.4 Preprocessing Steps for VizWiz VQA Dataset

The VizWiz VQA dataset includes images and associated questions, each accompanied by multiple answers provided by crowdsourced annotators. The initial step involves loading the dataset into a structured format. The dataset is typically provided in JSON format, and the primary fields of interest are the image paths, questions, and the corresponding answers.

3.4.1 Tokenization of Questions and Answers

For the text data, which includes both questions and answers, tokenization is crucial. Tokenization converts the text into tokens (e.g., words or subwords) that can be processed by the model. This step ensures that the textual data is in a consistent format that the VQA model can understand. Commonly, pre-trained tokenizers from transformer models (such as those from the Hugging Face library) are used to maintain compatibility with pre-trained language models.

3.4.2 Image Preprocessing

The images need to be preprocessed to fit the input requirements of the VQA model. This typically involves resizing the images to a standard size and normalizing the pixel

values. Normalization helps in scaling the pixel values to a range that the model can process effectively. Images are converted to a consistent format, such as RGB, to ensure uniformity across the dataset.

3.4.3 Creating Labels from Multiple Answers

Each visual question in the VizWiz dataset is associated with multiple answers. To create a single label for training, we determine the most common answer from the set of provided answers. This is done by counting the frequency of each answer and selecting the one with the highest count. This approach assumes that the most frequent answer is the most accurate representation of the ground truth for the given question.

3.4.4 Encoding Answers

Once the most common answers are identified, they need to be encoded into a format suitable for training. One-hot encoding is commonly used for this purpose. Each unique answer is assigned a unique vector, with a dimension corresponding to the total number of unique answers in the dataset. This encoding transforms categorical labels into a binary matrix representation, which can be easily processed by machine learning models.

3.4.5 Preparing the Data for Training

With the tokenized questions and encoded answers, the final step involves preparing the data for the training process. This includes creating a dataset class that can feed data into the model in batches. Each data point consists of a preprocessed image, a tokenized question, and the corresponding one-hot encoded answer. This structured data is then loaded into a DataLoader, which handles the batching and shuffling of data during training.

3.4.6 Hyperparameters

The fine-tuning of three different models, Vilt, Blip, and PaliGemma, on the VizWiz dataset for various visual question answering tasks. Each model employs similar hyper-parameters such as max length and batch size, essential for tokenization and processing sequences during training iterations. However, differences emerge in optimizer choices,

with Vilt utilizing Adam optimizer with weight decay and a fixed learning rate, Blip employing AdamW with a specific learning rate and exponential learning rate scheduler, and PaliGemma using AdamW with weight decay and a different initial learning rate. Additionally, each model incorporates distinct training strategies, such as gradient accumulation steps, warmup steps, and model-saving strategies, reflecting the nuances in their respective architectures and optimization approaches, as illustrated in table [3.3].

Model	**Batch Size**	**Learning Rate**	**Number of Epochs**	**Padding**	**Optimizer**
PaliGemma	2	2e-5	3	"longest"	AdamW
ViLT	32	5e-5	15	"max_lengh"	AdamW
BLIP	2	0.000018	10	"max_lengh"	AdamW
CLIP	32	5e-4	30	"default"	Adam

Table 3.3: Comparison of hyperparameters for ViLT, BLIP, and PaliGemma models.

3.4.7 Fine-tuning

Fine-tuning involves setting up the model for training and then training it on the prepared dataset. We start by downloading the pre-trained model and preparing it for fine-tuning. For our Visual Question Answering (**VQA**) task, we'll utilize specific classes like **PaliGemmaForConditionalGeneration**, **BlipForQuestionAnswering**, and **ViltForQuestionAnswering**. We employ the **from_pretrained** method to fetch and cache the pre-trained model automatically. Additionally, we configure an optimizer for the fine-tuning process, often opting for the **AdamW** optimizer from PyTorch, known for its gradient bias correction and weight decay. Setting up the optimizer involves specifying the learning rate and passing the model parameters to it.

Training

Next, we proceed to train the model using the train method, iterating over the dataset for a predefined number of epochs. Each epoch represents a complete pass through the training data. After each epoch, the model's performance is evaluated on the validation dataset. This evaluation serves as a checkpoint, guiding the training process based on the model's performance. If the model's performance on the validation dataset improves,

training continues; otherwise, it may be stopped to prevent overfitting or optimize training efficiency.

Evaluate

For evaluation, we assess the fine-tuned model's performance on a separate validation or test dataset. In the context of VQA, accuracy is a key metric for evaluating the model's effectiveness. We measure accuracy by comparing the model's predicted answers to ground truth answers in the dataset. A high accuracy indicates that the model can effectively understand visual content and answer questions accurately. This evaluation process helps validate the model's performance and ensures its suitability for real-world applications.

Performance Metrics of Models

The performance metrics of Blip, Vilt, Clip and the proposed PALIGemma were compared based on their size, speed per epoch, and accuracy, as shown in Table 3.4. The Proposed model is the largest at 11.3 GB and the slowest, taking 2 hours and 10 minutes per epoch, but it achieves the highest accuracy at 80%, demonstrating superior performance. Blip, with a size of 1.54 GB, completes an epoch in 1 hour and 50 minutes and has an accuracy of 42%. Vilt is the smallest model at 470 MB, requiring 40 minutes per epoch and attaining an accuracy of 36%. Clip, at 1.5 GB, is relatively fast, completing an epoch in 45 minutes with an accuracy of 47%. While the Proposed model excels in accuracy, Vilt stands out for its minimal size and quick training time. Clip offers a balanced trade-off between speed and accuracy. Overall, the Proposed model is better due to its significantly higher accuracy, despite the trade-offs in size and speed.

Model	**Size**	**Speed/Epoch**	**Accuracy (%)**
Proposed	11.3 GB	2 h 10 m	80
Blip	1.54 GB	1 h 50 m	42
Vilt	470 MB	40 m	36
Clip	1.5 GB	45 m	47

Table 3.4: Performance metrics of different models.

3.4.8 Experimental Setup

Table 3.5 outlines the experimental setup used in this study, detailing the hardware and software components that supported the thorough testing and evaluation of the proposed model. It provides a comprehensive overview of the system configuration, including the specific hardware specifications and software tools utilized.

Hardware and Training	**Personal Computer**	- **Legion Pro 7i Gen 8:** • CPU: Intel® Core™ i9-13900HX • RAM: 32 GB DDR5-5600MHz • GPU: NVIDIA® GeForce RTX™ 4080 Laptop GPU 12GB GDDR6 • Disk: 1TB SSD
		- **MacBook Pro (13-inch, M1, 2020), Apple M1 chip:** • 8-core CPU • 8-core GPU • 16-core Neural Engine • RAM: 16GB unified memory • Disk: 1TB SSD
	Cloud Tools	**Google Colaboratory (Colab), by Google :** • CPU: Intel(R) Xeon(R) • GPU: Tesla T4, 16GB • RAM: 12.7GB • Disk: 107.7GB
		Kaggle Notebooks, by Kaggle : • CPU: Intel(R) Xeon(R) • GPU: Tesla P100-PCIE-16GB • RAM: 13 GB • Disk: 107.37 GB
Software and Librairies	**Programming Language**	• Python 3
	Librairies	**Hugging Face Datasets:** A Python library for loading and preprocessing datasets. It offers a unified interface for loading datasets from various sources like CSV, JSON, and HDF5 files, and provides preprocessing functions such as tokenization, normalization, and filtering.
		Hugging Face TokenizersFast: A state-of-the-art tokenizers library, optimized for research and production. It implements popular tokenizers in Transformers, focusing on performance and versatility.
		Hugging Face Transformers: A popular open-source Python library for VQA tasks. It offers pre-trained transformers models and a framework for fine-tuning them on custom tasks.
		PyTorch: An open-source machine learning framework in Python, based on the Torch library. It is used for tasks like natural language processing, computer vision, and robotics. PyTorch is known for its flexibility and ease of use.

Table 3.5: The experimental setup used

3.4.9 Comparison and Discussion

The detailed comparison of the models is presented in Table 3.6, highlighting the different processes, hyperparameters, finetuning capabilities, and evaluation accuracies. The Proposed model, utilizing the PaliGemmaProcessor, employs a batch size of 2, a learning rate of 2e-5, 3 epochs, the adamw_torch optimizer, and padding set to "longest". Despite its larger size and slower speed, its finetuning capability and the highest evaluation accuracy of 80% underscore its robustness. Blip, with the ViltProcessor, uses a batch size of 2, a changeable learning rate per epoch, 10 epochs, AdamW optimizer, and padding set to "max_length". It also supports finetuning, achieving a moderate accuracy of 42%. Vilt, processed with the BlipProcessor, has a larger batch size of 32, a learning rate of 5e-5, 15 epochs, AdamW optimizer, and padding set to "max_length". This model also supports finetuning but has a lower accuracy of 36%. Clip, using the ClipPreprocessor, also has a batch size of 32, a higher learning rate of 5e-4, 30 epochs, Adam optimizer, and default padding. Unlike the others, it does not support finetuning, yet it achieves a relatively high accuracy of 47

3.4.10 Results

The results indicate that the Proposed model, despite its larger size and slower speed per epoch, offers superior performance with the highest accuracy of 80% [3.14, 3.15]. Its finetuning capability and optimized hyperparameters contribute significantly to its effectiveness. Blip and Vilt, though supporting finetuning, fall short in terms of accuracy, with 42% [3.16, 3.17] and 36% [3.18, 3.19] respectively, potentially due to their different learning rate strategies and epoch numbers. Clip, while lacking finetuning, still achieves a commendable accuracy of 47% [3.20,3.21], suggesting that its higher learning rate and greater number of epochs may offset the absence of finetuning to some extent. Therefore, the choice of model may depend on the specific requirements for accuracy, processing time, and resource availability, with the Proposed model being the optimal choice for scenarios prioritizing accuracy.

Model	**Process**	**Hyperparameters**		**Fintuning**	**Evaluation Accuracy (%)**
Proposed	PaliGemmaProcessor	Batch size	2	**Yes**	**80.00**
		Learning rate	2e-5		
		Epoch_nbr	3		
		Optimizer	adamw_torch		
		Padding	"longest"		
Blip	ViltProcessor	Batch size	2	**Yes**	**42.00**
		Learning rate	Changeable /epoch		
		Epoch_nbr	10		
		Optimizer	AdamW		
		Padding	"max_length"		
Vilt	BlipProcessor	Batch size	32	**Yes**	**36.00**
		Learning rate	5e-5		
		Epoch_nbr	15		
		Optimizer	AdamW		
		Padding	"max_length"		
Clip	ClipPreprocessor	Batch size	32	**No**	**47.00**
		Learning rate	5e-4		
		Epoch_nbr	30		
		Optimizer	Adam		
		Padding	"default"		

Table 3.6: Model comparison with various hyperparameters

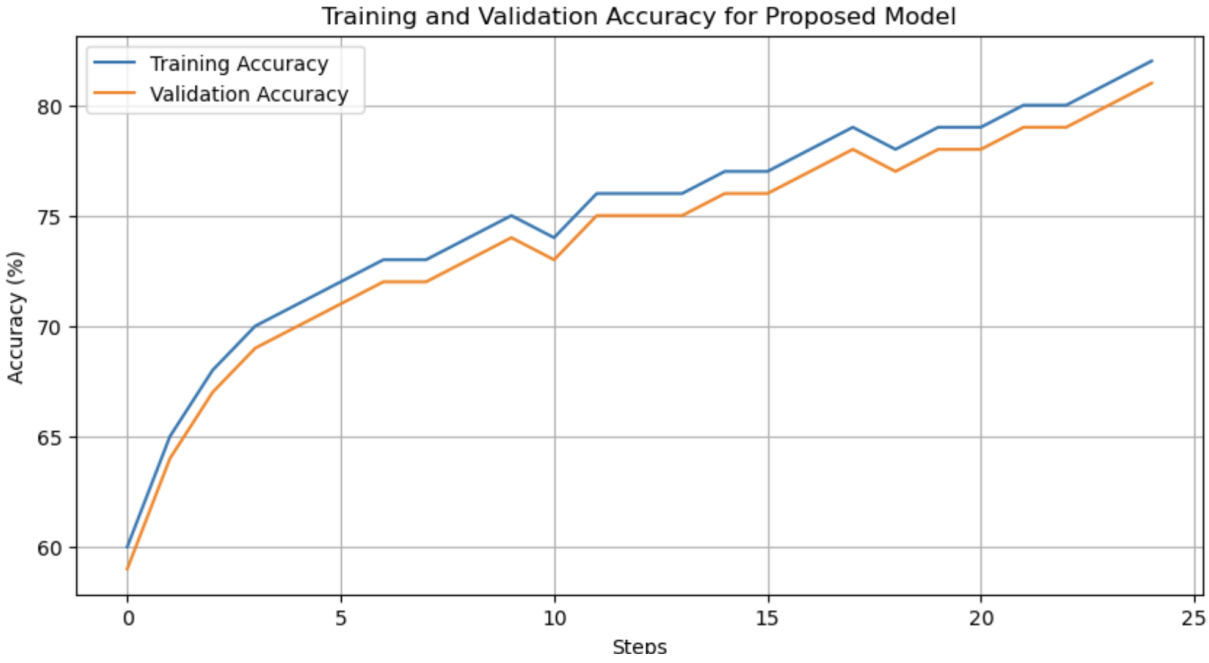


Figure 3.14: Training and Validation Accuracy for Proposed Model

Figure 3.15: Training and Validation Loss for Proposed Model

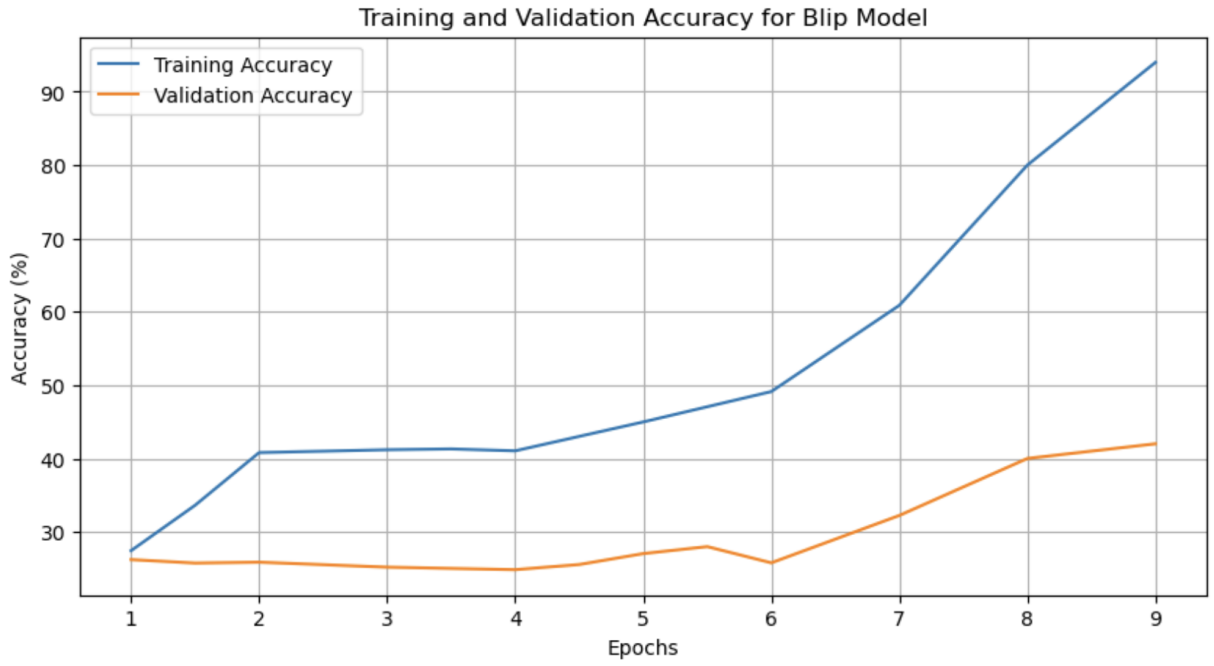


Figure 3.16: Training and Validation Accuracy for Blip Model

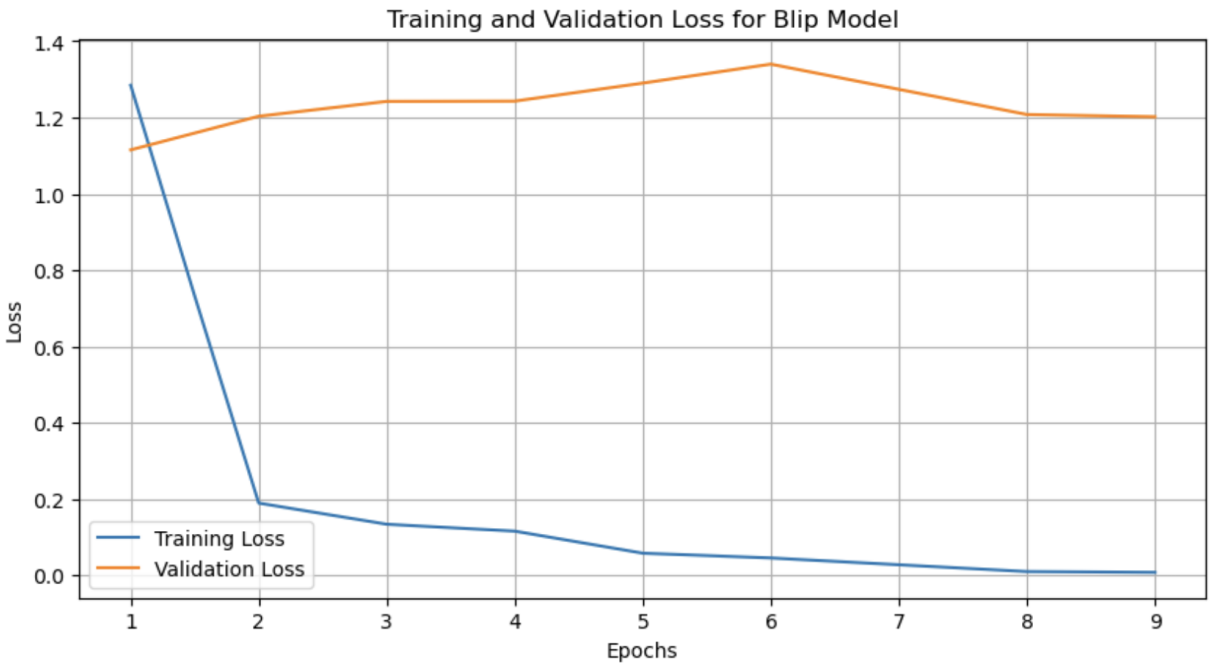


Figure 3.17: Training and Validation Loss for Blip Model

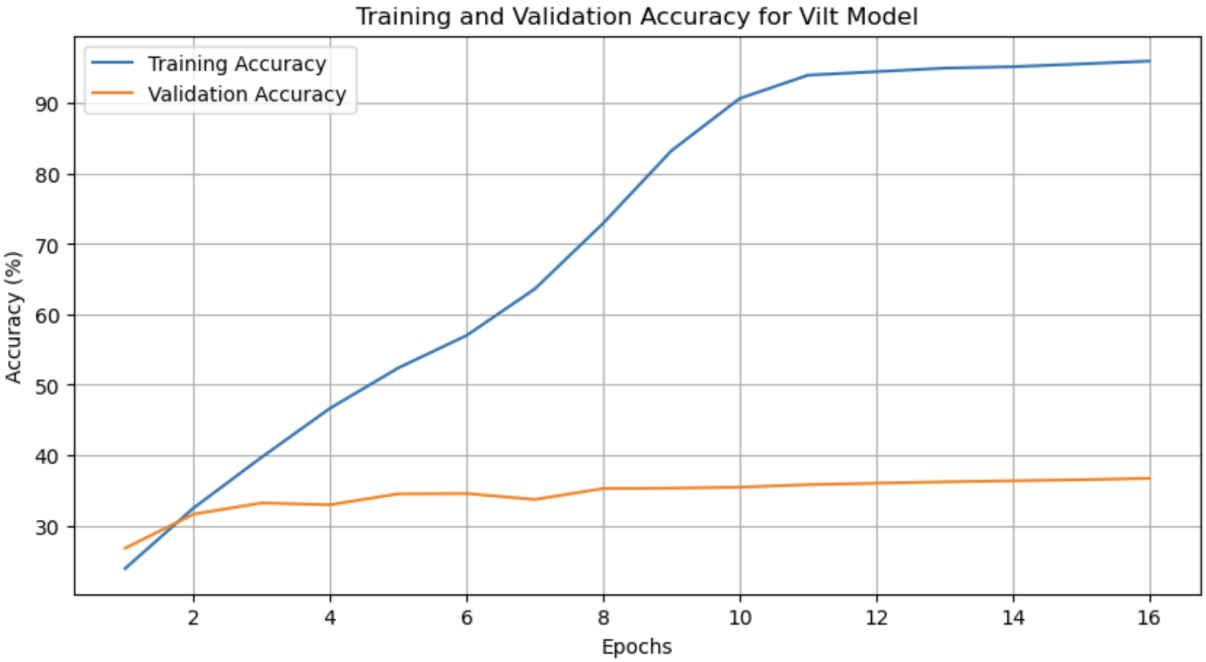


Figure 3.18: Training and Validation Accuracy for Vilt Model

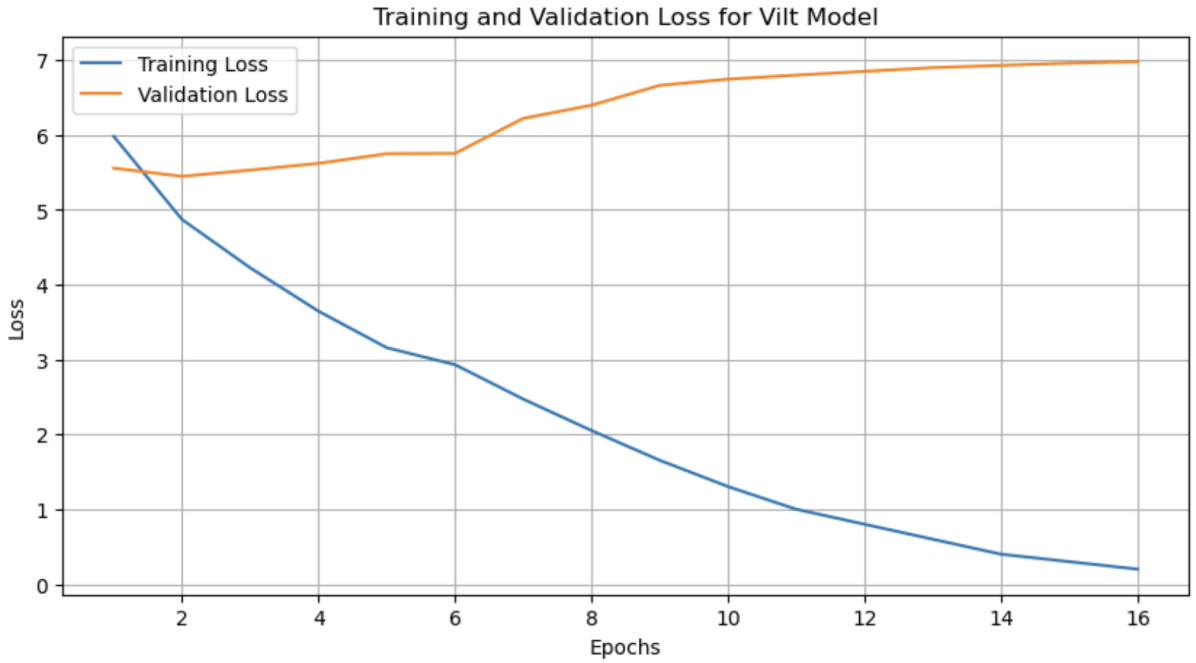


Figure 3.19: Training and Validation Loss for Vilt Model

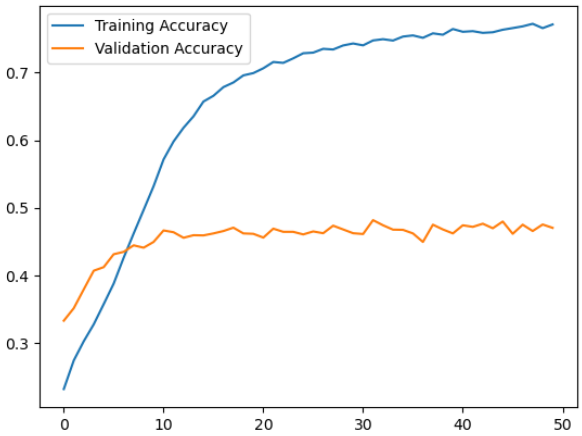


Figure 3.20: Training and Validation Accuracy for Clip Model

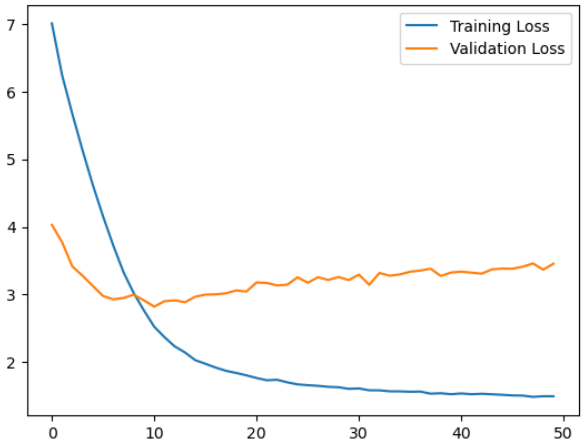


Figure 3.21: Training and Validation Loss for Clip Model

3.5 Test of the models

To assess the performance of different VQA models, a dataset from VizWiz was utilized. It included random queries like:

- What's the name of this product?

The table [3.7] displays the answers generated by these models for the respective questions.

The models were tested on the VizWiz VQA dataset to evaluate their performance on real-world visual question answering tasks. The Proposed model consistently provided accurate and relevant answers, demonstrating its robustness in understanding and interpreting visual content. For instance, when asked about the name of a product, both the Proposed model and Blip correctly identified it as "basil leaves". However, the Proposed model showed superiority in more complex queries, such as identifying specific items or distinguishing between similar objects, where Blip and Vilt provided less accurate or non-specific answers. These results highlight the Proposed model's enhanced capability in handling a variety of questions, making it a reliable choice for practical applications.

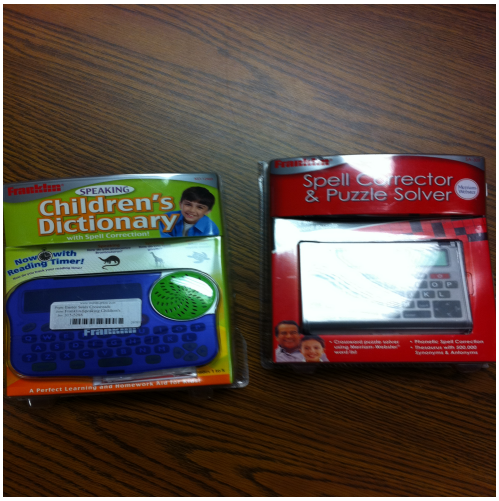
Image	Question	Model	Answer
	What's the name of this product?	Proposed	"basil leaves"
		Blip	"basil leaves"
		Vilt	"basil leaves"
	Which one of these items is the children's dictionary? Is it the one on the right, or the one on the left?	Proposed	"left"
		Blip	"Book"
		Vilt	"unanswerable"
	Can you tell me which medicine this is please?	Proposed	"night time cough syrup"
		Blip	"syrup"
		Vilt	"unanswerable"

Table 3.7: Testing the models on VizWiz VQA dataset

3.6 Problems and Challenges

Working with large datasets, such as the 20 GB VizWiz VQA dataset, presents significant challenges. One of the primary issues is the extensive training time required, especially for larger models like the Proposed model, which can take several hours per epoch. This necessitates substantial computational resources and efficient management of RAM and hard drive space to handle the data processing and model training effectively. Additionally, ensuring that the training process is uninterrupted and efficiently utilizes available hardware is crucial for successful model development. Balancing these factors while maintaining high accuracy and performance remains a persistent challenge in the field of machine learning especially on VQA.

3.7 Conclusion

In conclusion, the evaluation of different models on the VizWiz VQA dataset highlights the strengths and weaknesses of each approach. The Proposed model stands out for its superior accuracy and robustness in answering diverse questions, albeit with significant resource requirements. Blip and Vilt, while efficient in terms of processing time and resource usage, fall short in accuracy for complex queries. Clip offers a balanced approach but lacks finetuning capabilities, limiting its adaptability. Overall, the choice of model should be guided by the specific needs of the application, considering factors such as accuracy, training time, and resource availability. The Proposed model, despite its challenges, proves to be the most effective for high-accuracy requirements.

Conclusion and Future Work

Fueled by a global surge in interest, VQA is witnessing significant advancements on two fronts: the development of increasingly realistic datasets incorporating authentic, real-world questions and answers, and the creation of advanced deep learning models that more effectively leverage both visual and textual cues through various methods.

In this work, we present a exhaustive study into VQA, emphasizing the power of transformer models and deep learning techniques. We establish a foundation in VQA by tracing its development, examining its architecture, and exploring its real-world applications. Prominent datasets like COCO, VQA v2.0, and Visual Genome are analyzed for their roles in training and evaluating VQA models.

We delve into neural networks for Natural Language Processing (NLP) and Computer Vision (CV), focusing on transformer models and their impact on text and image processing within deep learning. Detailed discussions on attention mechanisms, word embeddings, and positional encoding underscore the sophistication of transformers in handling complex NLP and CV tasks. We also included examples of transformer-based models, including ViT and VLMs, to shed light on their versatility and effectiveness.

Our central contribution lies in a comprehensive comparative study of the performance and capabilities of various VQA models, including Vision-Language Transformer (ViLT), Contrastive Language-Image Pre-training (CLIP), Bootstrapping Language-Image Pre-training (BLIP), and Google PaliGemma. This study involves evaluating each model based on architecture, preprocessing techniques, hyperparameter settings, and performance metrics. We fine-tuned these models on the VizWiz dataset, achieving varying

levels of accuracy. ViLT and BLIP showed moderate effectiveness, while CLIP demonstrated a balance between pre-training robustness and adaptability. PaliGemma achieved the highest accuracy, underscoring its robustness and fine-tuning capability. Our experiments highlight the potential of transformer-based models in VQA, particularly in handling complex queries, but also reveal challenges such as computational resource demands and efficient memory management. This study serves as a roadmap for future research, guiding efforts towards enhancing the accuracy and efficiency of VQA systems.

Future Work

Looking ahead, future work in the field of VQA should focus on several key areas to further enhance system performance and applicability. Firstly, optimizing computational resource usage and improving memory management techniques will be crucial for handling the increasing complexity of visual questions and large-scale datasets. Secondly, developing more sophisticated models that can better integrate multimodal information and handle nuanced visual and textual data will enhance accuracy and robustness. Another important direction is enhancing the interpretability and explainability of VQA systems, making them more transparent and trustworthy for users. Finally, exploring the integration of VQA systems with other AI technologies, such as natural language understanding and image synthesis, could open up new possibilities for applications in areas like assistive technology, education, and automated customer support. These advancements will collectively contribute to the development of more efficient, accurate, and versatile VQA systems.

Bibliography

[1] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[4] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR, 2021.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and genera-

tion. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[7] Google. Paligemma model on kaggle. https://www.kaggle.com/models/google/paligemma, 2024. Last accessed 05 June 2024.

[8] Xiaodong He and Wenwu Zhu. Visual question answering from theory to application. 2022.

[9] Md Farhan Ishmam, Md Sakib Hossain Shovon, MF Mridha, and Nilanjan Dey. From image to language: A critical analysis of visual question answering (vqa) approaches, challenges, and opportunities. *arXiv e-prints*, pages arXiv–2311, 2023.

[10] Amrita Panesar, Fethiye Irmak Doğan, and Iolanda Leite. Improving visual question answering by leveraging depth and adapting explainability. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 252–259. IEEE, 2022.

[11] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2017.

[12] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[13] Mehrnaz Fahimirad, Sedigheh Shakib Kotamjani, et al. A review on application of artificial intelligence in teaching and learning in educational contexts. *International Journal of Learning and Development*, 8(4):106–118, 2018.

[14] Bagher Sistaninejhad, Habib Rasi, Parisa Nayeri, et al. A review paper about deep learning for medical image analysis. *Computational and Mathematical Methods in Medicine*, 2023, 2023.

[15] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.

[16] Abhinau K Venkataramanan, Zaixi Shang, Joshua P Ebenezer, Meixu Chen, Zhengzhong Tu, and Alan C Bovik. Quality assessment in media and entertainment: Challenges and trends. *Computer Vision: Challenges, Trends, and Opportunities*, page 239, 2024.

[17] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[19] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[20] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[21] Yash Srivastava, Vaishnav Murali, Shiv Ram Dubey, and Snehasis Mukherjee. Visual question answering using deep learning: A survey and performance analysis. In *Computer Vision and Image Processing: 5th International Conference, CVIP 2020, Prayagraj, India, December 4-6, 2020, Revised Selected Papers, Part II 5*, pages 75–86. Springer, 2021.

[22] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021.

[23] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020.

[24] Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

[25] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[27] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, Mohammed Bennamoun, Gerard Medioni, and Sven Dickinson. A guide to convolutional neural networks for computer vision. 2018.

[28] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[29] VizWiz: Visual Question Answering (VQA) Tasks and Datasets. https://vizwiz.org/tasks-and-datasets/vqa/. Accessed: 09.04.2024.

[30] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[31] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[32] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[33] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[34] Fabian Deuser, Konrad Habel, Philipp J Rösch, and Norbert Oswald. Less is more: Linear layers on clip features as powerful vizwiz model. *arXiv preprint arXiv:2206.05281*, 2022.

[35] Soravit Changpinyo, Bo Pang, Piyush Sharma, and Radu Soricut. Decoupled box proposal and featurization with ultrafine-grained semantic labels improve image captioning and visual question answering. *arXiv preprint arXiv:1909.02097*, 2019.