Visual Robustness Analysis and Caption Ranking for Zero-shot Visual Question Answering

Ishmam Tashdeed, 180041105

Md. Farhan Ishmam, 180041120

Talukder Asir Saadat, 180041127

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
May, 2023

Visual Robustness Analysis and Caption Ranking for Zero-shot Visual Question Answering

by

Ishmam Tashdeed, 180041105
Md. Farhan Ishmam, 180041120
Talukder Asir Saadat, 180041127

Supervisor
Md. Hamjajul Ashmafee
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology

BACHELOR OF SCIENCE
IN
COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Board Bazar, Gazipur-1704, Bangladesh.
May, 2023

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Ishmam Tashdeed**, Student ID: 180041105, **Md Farhan Ishmam**, Student ID: 180041120 and **Talukder Asir Saadat**, Student ID: 180041127, under the supervision of **Md. Hamjajul Ashmafee**, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Md. Hamjajul Ashmafee
Assistant Professor
Department of CSE
Islamic University of Technology (IUT)
Date: May 27, 2023

Ishmam Tashdeed
Student No.: 180041105
Date: May 27, 2023

Md. Farhan Ishmam
Student No.: 180041120
Date: May 27, 2023

Talukder Asir Saadat
Student No.: 180041127
Date: May 27, 2023

Dedicated to our parents and loved ones

Table of Contents

TABLE OF CONTENTS

TABLE OF CONTENTS

List of Figures

List of Tables

Page

List of Abbreviations

VQA	Visual Question Answering
ZS-VQA	Zero-Shot Visual Question Answering
VRE	Visual Robustness Error
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
FFN	Feed Forward Network
ITM	Image-Text Matching
CA	Cross-Attention
MLP	Multilayer Perceptron
VLP	Vision-and-Language Pretraining
ViLT	Vision and Language Transformer
GIT	Generative Image-to-text Transformer
LM	Language Modeling
VLM	Visual Language Model
LLM	Large Language Model
VLKD	Vision Language Knowledge Distillation
BLIP	Bootstrapping Language-Image Pre-training
MED	Mixture of Encoder-Decoder
CLIP	Contrastive Language–Image Pre-training
GPT	Generative Pre-trained Transformer
BQD	Basic Questions Dataset
CARETS	Consistency And Robustness Evaluative Test Suite
BLEU	BiLingual Evaluation Understudy
QIP	Question Invariant Prompting
TAP-C	Template-Answer Prompt then CLIP
PNP	Plug and Play
HSV	Hue Saturation Value

Acknowledgment

We would like to express our gratitude towards the IUT authority for granting us the fund and providing the assistance required to implement our proposed system. We are indebted to our supervisor and co-supervisors, Md. Hamjajul Ashmafee, Dr. Md. Azam Hossain, and Mohammad Ishrak Abedin, for providing us with insightful knowledge and guiding us at every stage of our journey. We are also grateful to the head of the CSE department, Dr. Abu Raihan Mostofa Kamal for establishing the Deep Learning Research Lab. Finally, we would like to express our heartiest appreciation towards our family members for their continuous support, motivation, suggestions, and help, without which we could not have achieved the scale of implementation that we have achieved.

Abstract

The domain of Visual Question Answering (VQA) is an important cornerstone for the understanding of the combination of both visual and textual realms. Robustness is the ability of a model to resist adversarial attacks and has been a research interest in VQA. Although the linguistic robustness of VQA methods has been a common field of interest, there has yet to be any significant work on visual robustness. We present a framework that focuses on challenging the visual robustness of multiple VQA models by applying realistic visual corruption effects to the VQA datasets. We also introduce several metrics to quantify a model's robustness along with an aggregated metric – Visual Robustness Error (VRE) that provides a single value corresponding to the model's error in dealing with corrupted data. We observe that the high accuracy achieved by current VQA models does not necessarily translate to a high robustness score. We intend our method to be used for evaluating the robustness of various VQA methods and evaluating the strength of various corruption effects.

Recent works regarding zero-shot VQA have been accelerated as large language models (LLMs) have been improving in both quality and quantity. Existing methods took advantage of these large language models and using network interpretation techniques as an interface, made a modular framework for zero-shot VQA. We propose a method to improve upon its predecessors by adding a caption ranking method to generate higher-quality captions. The standard approaches generate a certain number of captions and subsequently pass them to the next module for predictions. The model's inference time depends on the number of captions used. By ranking the captions and picking the top-ranked ones, we are able to reduce the number of captions required from 100 to 5 and decrease the inference time by 22.25% sacrificing only 2.84% accuracy.

Chapter 1

Introduction

In this chapter, We provide an inaugural outline of our work starting with an introduction to the research area of Visual Question Answering (VQA) which deals with answering a question based on an image. We explore the basic concepts behind visual robustness and its applications in the domain of VQA. We further discuss the notion of Zero-Shot setting in VQA and present a precise exploration of the existing research conducted in the field of zero-shot VQA along with its utilization in real-life scenarios. We formulate our problem statement, mention the research challenges we faced throughout the journey, and follow up with our research objectives and contributions. Finally, we end the chapter with the organization of the rest of the thesis.

1.1 Overview

Visual Question Answering (VQA) [21] is a monumental task in the field of multi-modal research for vision-language reasoning and comprehension. VQA requires domain expertise in both image processing and natural language processing. A VQA system can be described as a deep learning system that takes an image-question pair as the input and generates an answer as the output that satisfies the question with the image given as context [1, 15, 22]. In essence, VQA is an extension to **contexual question-answering** [23] where a paragraph is passed as context to the question. With the advent of advanced vision and language processing methods, various efforts have been made to build the vision-language pre-trained models for capturing the alignment between vision and language [4, 24–26]. We use these methods for VQA as they grasp the relationship between the image provided as context and the question.

VQA methods are crucial in solving many *real-life* problems [27]. VQA systems can also be integrated into assistive technologies such as screen readers, allowing visually impaired people to easily access and understand image and video content [28, 29]. An emerging application of VQA is image-based chatbots to provide quick and accu-

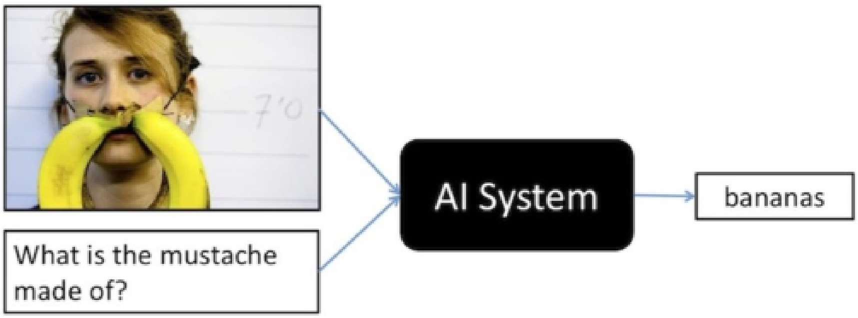


Figure 1.1: Abstraction of VQA [1]

rate responses to product or service-related questions by customers [30].

VQA as a research problem aims to appropriately answer any type of question given any form of an image as context, and it should not be limited by certain image, question, or answer constraints. Zero-shot VQA (ZS-VQA) is a sub-domain of VQA that aims at VQA models predicting *out-of-distribution* data [31]. Defining *out-of-distribution* data is subjective and hence, ZS-VQA alternatively works with models that are not *specifically trained* in VQA. Zero-shot VQA primarily aims at the problem of generalized VQA and utilization of data from an external knowledge source.

1.2 Visual Robustness

Robustness is the ability of a model to resist adversarial attacks and still provide expected results. As VQA is a multi-modal task, it is susceptible to adversarial disturbances of both modalities. These perturbations can be small changes in questions like changing some words with their synonyms or antonyms or in images through visual noise [7]. Although the linguistic robustness of VQA methods has been a common field of interest [6, 8], there has yet to be any significant work on exploring visual robustness. A VQA method would be considered *visually robust* if it can provide correct or expected answers even after the visual input (the image) has been degraded to a certain degree with specific perturbations that challenge the model.

Adversarial attacks first encountered by Szegedy et al. [32] are considered to be a shortcoming of modern deep learning methods. The weakness was initially uncovered in the context of image classification [7] which led to the phenomena being observed in various sub-tasks of both computer vision and natural language processing [6–8]. The effects of adversarial perturbations were soon noticed in language processing tasks [33]. Thus, it is no surprise that multi-modal tasks such as VQA are vulnerable to such attacks. Hence, assessing the robustness of VQA methods to such adversarial attacks is essential.

1.3 Zero-shot VQA (ZS-VQA)

Zero-shot learning is a well-defined challenge in machine learning, where the task is to classify samples into classes that were not encountered during training. In the context of Visual Question Answering (VQA), the existing methods typically rely on datasets consisting of question, image, and answer combinations, covering various question types across different objects and scenarios [34]. However, it is impractical to have a finite set of examples that encompass the immense diversity of the real world, which a truly ideal VQA system should be able to handle and understand. So by design, the VQA task is zero-shot from its inception. Due to modern VQA methods using large language models (LLMs) pre-trained on vision-language tasks [11, 12], finding out-of-distribution examples could be very difficult. One downside to training VQA methods using established datasets is that the loss functions force the models to output the most frequent answers and due to inherent biases in the dataset, they often achieve high accuracy scores [35].

A different approach to enable zero-shot capabilities in a method involves achieving the task without explicitly training a model on that specific task [11, 12] and the approach has been quite successful in tasks such as image classification [36, 37], image captioning [38], digit recognition [39], harmful content detection [39], and many more. For instance, CLIP [9], which is trained using contrastive loss to match images and textual captions, can be utilized in a zero-shot setting of other problem domains. Recently, CLIP has been used for ZS-VQA [10, 11] and sparked a recent trend of leveraging large pre-trained language models and using them for zero-shot VQA [12]. Exploring zero-shot settings is therefore crucial for advancing VQA methods and capitalizing on the significant performance gains offered by these pre-trained models.

1.4 Problem Statement

Through our literature review, we uncovered a lack of methods that determines the robustness of VQA methods. Although some methods exist, most of them focus on the textual modality. In the current literature, there is an absence of a comprehensive framework that tests the visual robustness of VQA methods by applying multitudes of specific visual adversarial perturbations. This is a possible area of contribution and can be cemented as a vital test whenever a model is being evaluated.

Furthermore, the ability of vast generalization and the capability of taking advantage of new pre-trained large language models or image encoding methods makes it very lucrative to explore the zero-shot setting for VQA tasks. But this is an emerging field and very few methods exist that leverage these capabilities. One of the novel methods that have propelled the current state-of-the-art for zero-shot VQA uses captions generated from attending different patches of the image [12]. We could potentially improve upon this architecture by using a mechanism for ranking the captions and only using high-quality captions which would enable the model to generate more potent captions, in turn providing us with high-quality answers and reducing the inference time. Thus, a feasible contribution could be to devise a novel method for zero-shot VQA, by ranking the generated captions by leveraging image-text similarity scores obtained from methods such as [9].

So, our problem statement can be summarized as: *"Creating a comprehensive and modular framework for assessing the visual robustness of a VQA method and devising a novel method for zero-shot VQA through ranking generated captions."*

1.5 Research Challenges

In this section, we examine the difficulties encountered by researchers in the fields of VQA and ZS-VQA. Given the fast-paced advancement of both domains, it is possible that these challenges may be resolved by the time our work is completed.

1.5.1 Visual Robustness for VQA

There are some challenges that make the task of assessing the visual robustness of VQA methods quite difficult. The first issue is to ascertain which visual noise is going to be effective for determining if a model is *robust*. We cannot assess the models just by adding any type of perturbations. It has to test how much the model *understands* the image and the question. So the noise added to the images should keep the semantic elements of the image intact and inquire up to what level the model can provide expected answers. Also, adding unrealistic adversarial noise will not benefit the model in real-life scenarios. If the model is robust against unrealistic noise but fails to generalize against realistic perturbations then it is less useful after deployment. Images are subject to many different situations that may produce less ideal images. The datasets [1, 15] used to train these state-of-the-art models do not contain such visually corrupted images. Some examples of image corruption could be: overexposure, low brightness, motion blur, defocused blur, image compression, pixelation, and many more [7]. Any method that judges the robustness of VQA methods should assess the models on these criteria as well.

1.5.2 Zero-shot VQA

VQA is a challenging task in and of itself as it is a multi-modal problem. Recent trends have shown that methods using large language models and large image encoders in a vision-language pre-training setup usually do better. But this comes with the overhead of training these large vision-language models. Also, as these are trained using the standard **VQAv2** [15] dataset most of the time, they do not generalize well. The inherent biases in the dataset encourage the model to answer the most frequent answers which can be disastrous for certain situations. This paradigm encourages the use of zero-shot VQA methods for their excellent generalization abilities. But zero-shot methods have some hurdles of their own. Current methods [37] mostly use pre-trained large image and language processing models for VQA but these models are unimodal. Some methods use natural language as a connector between the two modalities [12]. Other methods use the encoder-based methods such as representations learned from contrastive loss using a model like CLIP [10, 11]. It is quite challenging to come up with effective methods to bridge this gap of modalities. Any successful zero-shot VQA method should ensure an effective way to interface between the visual and language modalities.

1.6 Research Objectives

Our research objectives encompass both VQA and ZS-VQA and can be summarized in the following key points:

- Define a comprehensive framework that can assess the visual robustness of a VQA method along with necessary evaluation metrics.
- Compare realistic visual corruptions affecting current VQA methods.
- Design a novel zero-shot VQA module to reduce inference time.
- Optimization of interfacing between the two modalities of image and text in Zero shot settings.

1.7 Contribution

We contribute to two important aspects of Visual Question Answering (VQA): zero-shot VQA and visual robustness. In zero-shot VQA, we optimize inference time by reducing the number of captions required while maintaining high accuracy. For visual robustness, we propose techniques to evaluate and quantify models' ability to handle adversarial disturbances in both vision and language. Our contributions enhance the efficiency and resilience of VQA models in real-world applications.

Our contributions to Visual Robustness for VQA methods are:

1. **Visual Robustness Evaluation**: We develop a comprehensive framework to assess the visual robustness of multiple VQA models. By applying realistic perturbations to the VQAv2 dataset, we simulate real-world scenarios where visual inputs may be corrupted or distorted. This evaluation framework provides insights into the models' vulnerabilities and their ability to generate accurate predictions in the presence of visual disturbances.

2. **Robustness Metrics**: To quantify the models' robustness, we introduce several metrics that capture different aspects of their performance. These metrics allow us to measure the models' error rates and performance degradation when faced with various types of corruption. Our approach provides a more nuanced understanding of the models' limitations and helps identify areas for improvement.

3. **Visual Robustness Error (VRE)**: As an aggregated metric, VRE provides a single value that summarizes the model's error in handling corrupted data. By comparing VRE scores across different models, we gain insights into their relative performance and overall robustness. This metric serves as a useful benchmark for evaluating the stability of various VQA methods.

Furthermore, our contributions to Zero-shot VQA are:

1. **Caption Ranking Method**: We introduce a caption ranking technique to enhance the quality of captions generated in VQA systems. By assigning ranks to different captions based on their relevance and accuracy, we improve the overall performance of the models in generating high-quality captions.

2. **Handling Adversarial Disturbances**: Our approach takes into account the susceptibility of VQA models to adversarial disturbances in both vision and language modalities. By considering these potential disturbances during the caption generation process, we aim to improve the robustness and reliability of the models in handling diverse visual and linguistic inputs.

3. **Inference Time Reduction**: We address the high inference time of zero-shot models by optimizing the caption generation process. Through caption ranking and selecting the top-ranked captions, we significantly reduce the number of captions required from 100 to 5. This reduction in captions leads to a substantial decrease in inference time by 22.25% while maintaining a minimal sacrifice of only 2.84% in accuracy.

In conclusion, our work addresses the challenges of zero-shot VQA and visual robustness in the field of Visual Question Answering. By optimizing inference time and evaluating models' resilience to adversarial disturbances, we contribute to the advancement of more efficient and reliable VQA systems. These contributions pave the way for enhanced performance and practical application of VQA models in various domains.

1.8 Organization of the Thesis

Chapter 2 of this thesis delves into an extensive review of the existing literature on Visual Question Answering (VQA), visual robustness, and zero-shot VQA. This review provides a comprehensive understanding of the current state of research in these areas, highlighting the key findings, methodologies, and challenges encountered by researchers.

Moving forward, **Chapter 3** focuses specifically on the concept of visual robustness. It explores various techniques and approaches employed to enhance the robustness of VQA models when faced with perturbations or adversarial disturbances in the visual and textual input. The chapter examines different metrics and evaluation methods used to quantify the resilience of VQA models and presents insights into the vulnerabilities and limitations observed in current approaches.

In the subsequent chapter, **Chapter 4**, the thesis delves into the intriguing domain of zero-shot VQA. It explores the unique challenges posed by zero-shot settings, where the model is expected to answer questions about objects or scenes it has never encountered during training. The chapter investigates various strategies and techniques employed to enable zero-shot VQA. It discusses the limitations and potential avenues for improvement in this emerging field.

Finally, **Chapter 5** draws upon the insights and findings from the previous chapters to present a set of conclusions. These conclusions summarize the main contributions of the thesis, highlighting the advancements made in the domains of visual robustness and zero-shot VQA.

Chapter 2

Literature Review

In this chapter, we start by discussing the progression of VQA methods from early approaches to modern methods. We bring out the typical methods for robustness analysis in current machine learning and deep learning models followed by pointing out the existing methods for performing zero-shot VQA which contains discussions about how the zero-shot setting differs from regular VQA methods, the usage of vision-language pre-training, and adapting large language models for zero-shot VQA. This chapter focuses on several VQA models or methods and talks about their apparent contributions and limitations. Afterward, we study existing robustness testing methods for VQA and conclude the chapter with a discussion on existing zero-shot VQA methods.

2.1 Evolution of VQA Architectures and Robustness

Visual Question Answering [21] is one of the most challenging tasks being researched in multimodal learning. Excelling in this task requires a greater understanding of visual elements in the given context i.e. image or video, as well as processing the given question. Works such as [40–44] paved the way for further research as they constantly improved upon the accuracy score. But these methods contained inherent biases either from the distribution of the datasets [45] or from the modalities such as the question given [40, 43, 46, 47]. This frequently resulted in models answering correctly but for the wrong reasons.

Visual Question Answering has always been a unique problem domain as it combines the domains of Computer Vision and Natural Language Processing. The general VQA problem consists of answering any question with an image given as the context. The problem domain of VQA is similar to context-based textual question answering as seen in and can be thought of as an extension of contextual QA. By the end of the last decade, the field of visual question answering experienced rapid growth— primarily due to the advent of revolutionary architectures like transformers in processing

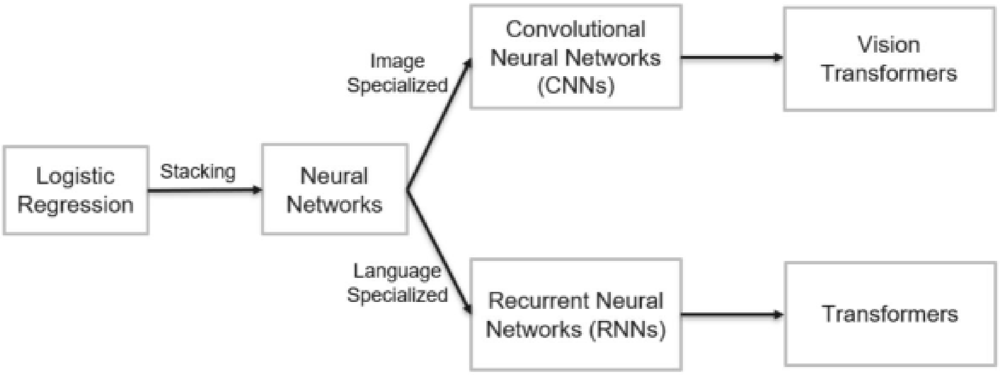


Figure 2.1: Evolution of VQA core architectures

sequential data and outperforming Recurrent Neural Networks (RNNs) and their variants. Recently, we have seen Vision Transformers outperforming Convolutional Neural Networks (CNNs) resulting in the creation of completely transformer-based architectures like ViLT [4] in VQA.

As VQA is a multimodal task, the model needs to perform inferences from images and textual data. [6] explored the robustness of a model in dealing with adversarial attacks which can target the question and/or the image. While the robustness of the textual sub-model has been thoroughly explored by experimenting with the questions fed to the model, there has been no analysis of the robustness of the visual sub-model by experimenting with the images given as context. In this paper, we will delve into the model's robustness in dealing with visual data by performing standard image processing experiments on common datasets and evaluating the performance of the model on the transformed datasets. Since the image serves as a context to the question, any substantial change to the image will affect the model's predicted outcome. We performed a series of image processing operations on a standard VQA dataset.

Adversarial attacks are considered to be a shortcoming of modern deep learning methods. The weakness was initially uncovered in the context of image classification which led to the phenomena being observed in various sub-tasks of both computer vision and natural language processing. The effects of adversarial perturbations were soon noticed in language processing tasks. Thus, it is no surprise that multimodal tasks such as Visual Question Answering (VQA) are vulnerable to such attacks. So, assessing the robustness of VQA methods to such adversarial attacks is essential.

2.2 Standard Robustness Analysis Methods

Robustness analysis is the process of evaluating how well a machine learning model performs under different conditions. This can include evaluating the model's performance on different datasets, with different types of input data, or under different operating conditions. The goal of robustness analysis is to identify potential weaknesses in the model and to improve its performance in the face of these challenges. There are several different approaches to performing robustness analysis, depending on the specific needs of the model and the application it is being used for. Some common techniques include:

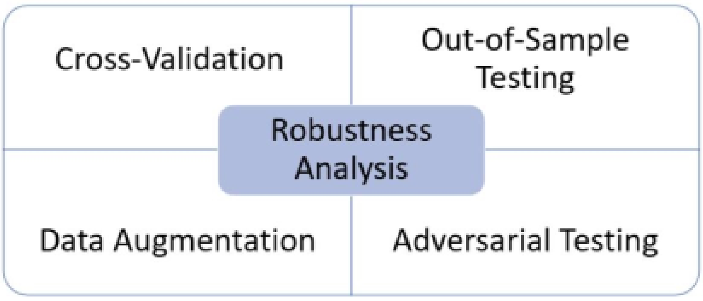


Figure 2.2: Standard Methods of Robustness Analysis

1. **Cross-validation:** Simplest form of a machine learning model's performance evaluation which involves dividing the training dataset into multiple smaller datasets and training the model on each of these datasets in turn. This allows you to assess the model's performance on different subsets of the data and can help identify any biases or overfitting. Cross-validation is loosely associated with robustness testing and has been widely integrated as an essential step for hyperparameter tuning in any ML pipeline.

2. **Out-of-sample testing:** This involves evaluating the model's performance on a dataset that it has not seen during training. This can help identify any issues with the model's generalization ability. The issue of testing out-of-sample data has been tacked by zero-shot models and has been typically excluded while evaluating the robustness of VQA models.

3. **Data augmentation:** This involves creating additional training examples by applying various transformations to the existing training data. This can help the model learn to be more robust to changes in the data distribution and can improve its performance on real-world data. We are relying on data augmentation to test the visual robustness of various VQA models.

4. **Adversarial testing:** This involves generating input data that is specifically designed to trick the model into making incorrect predictions. This can help identify any vulnerabilities in the model's decision-making process and can lead to improvements in its robustness. The adversarial testing can be done either by designing adversarial attacks based on the model's architecture, also known as white-box attacks, or based on only the inputs and outputs of the model, also known as black-box attacks. Due to the complexity of designing adversarial attacks for a variety of VQA architectural types, we will not be covering this class of robustness testing in our work.

Robustness analysis is an important aspect of building effective VQA models, as these models need to be able to handle a wide range of input data and operate under different conditions. Before diving deep into the robustness of the VQA models, we shall first look at various types of VQA models, the standard methodologies in VQA along with the datasets used by the VQA models.

2.3 Standard Zero-Shot VQA Methods

2.3.1 Traditional methods for VQA

At the inception of this field, the task of VQA was completed by processing the image and textual inputs separately in two streams and then conjoining them before producing an answer as seen on fig-2.5 [1, 15, 48, 49]. Soon, more complex methods concerning attending to different regions of the inputs became popular along with methods leveraging knowledge graphs and cross-attending features. Some methods employing these are seen in [31, 48–50]. But as more modern methods of language processing were established, these methods also changed. Specifically, the massive performance gained after the advancements made with large language models helped pave the way for modern VQA methodology.

2.3.2 Vision-Language Pre-training

This is a novel and popular research direction in the field of multi-modal image-text understanding. Various vision-language pre-training tasks have been proposed, including image-conditioned language modeling [13, 14]. We also see masked language modeling methods such as [24, 51, 52].

After the success of Radford *et al.* [9], learning the relationship between image-text modalities through contrastive loss has gotten extremely popular. [53, 54] exhibit zero-shot capabilities after pre-training on image-language tasks. Fig-2.3 shows the

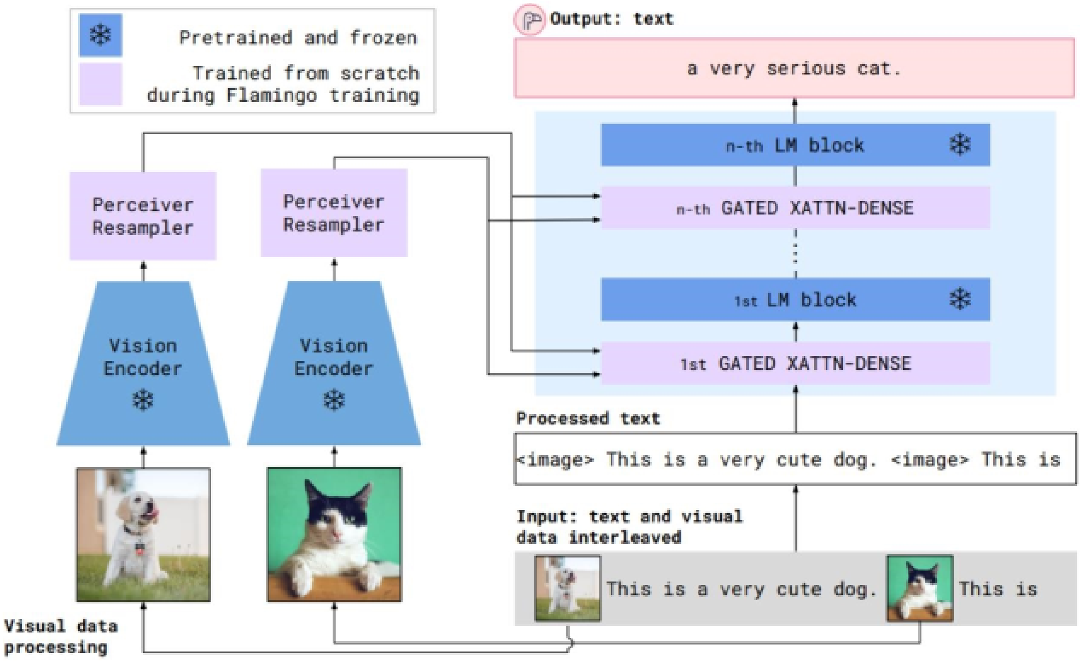


Figure 2.3: An overview of the Flamingo model [13] using Vision-Language Pre-training.

Flamingo models are a group of visual language models (VLMs) that can process free-form text as output and accept visual data mixed with text as input [13].

2.3.3 Adapting pre-trained LMs for zero-shot VQA

To integrate visual information into pre-trained language models, many current approaches involve conducting additional training that combines images and text. Tsimpoukelli et al. [14] take a different approach by training the vision encoder while keeping the large pre-trained language model frozen. This allows the language model to retain its knowledge in question answering, as illustrated in fig-2.4. This strategy enables the fusion of vision and language without sacrificing the expertise of the pre-trained language model in handling textual data.

Different approaches have been proposed to incorporate visual information into pre-trained language models. Tsimpoukelli et al. [14] adopt a strategy where the vision encoder's output is used as prompts alongside the frozen language model. Jin et al. [55] fine-tune the pre-trained language model using prefix language modeling and masked language modeling objectives. VLKD [56] leverages CLIP as a teacher model during fine-tuning to distill multi-modal knowledge. Aylarac et al. [13] introduce additional layers to both the pre-trained vision model and the language model, training them on a large-scale dataset of image-text pairs. Another approach explores training vision-language models on synthetic VQA examples generated from captions [57,58]. These various methods contribute to the advancement of vision-language integration

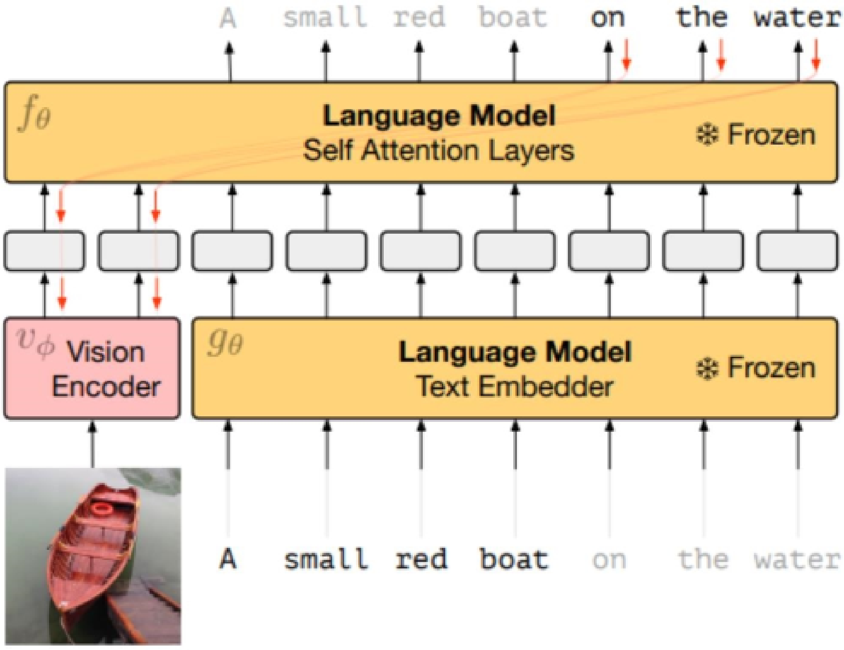


Figure 2.4: The vision encoder is trained using gradients across the self-attention layers of a FROZEN [14] language model demonstrating how FROZEN learns multi-modal features by maintaining the frozen state of the massive language models.

and offer promising avenues for zero-shot VQA tasks.

Utilizing big language models that have already been trained and image captions to provide a summary of the image is a straightforward but incredibly efficient unique way for zero-shot VQA. PICa adopts GPT-3's [59] for zero-shot VQA and turns an image into a single caption. Tiong et al. [12] cite this work and suggest the use of question-guided captions. In the parts that follow, we will go into further detail on this.

2.4 Visual Question Answering (VQA)

The methods of VQA experienced rapid development throughout the years and have been approached in a multitude of ways. The standard approach can be broken down into three key phases: feature extraction, feature conjugation, and answer generation. For many years, VQA has been treated as a multimodal task where question and image feature extraction has been done separately, followed by some form of feature conjugation which combines the two modalities. Image feature extraction primarily relied on standard computer vision models like LeNet [16], AlexNet [60], VGG [61], ResNet [62], and InceptionNet [63] - just a few examples fueled by the ImageNet [64] competition.

Usually, a pre-trained image model is used as a backbone network, which can provide a good image feature vector. VQA models primarily rely on ResNet and VGG

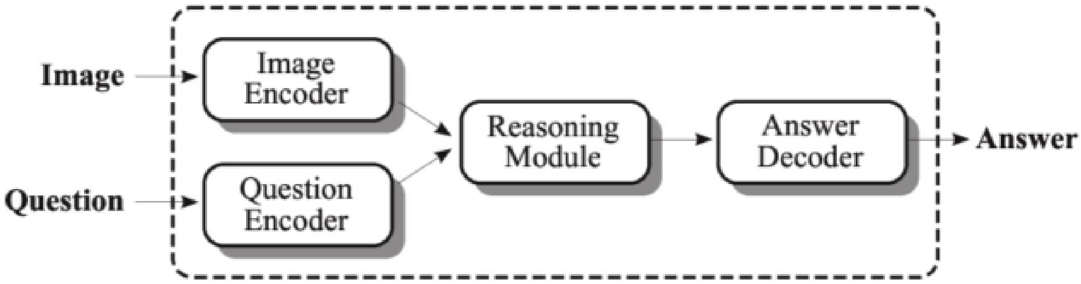


Figure 2.5: Standard architecture of VQA models using Multi-Modal Approach

as the backbone networks. After the advent of transformers, vision transformers were introduced in [65] breaking down the image into 16x16 image patches and passing it through a transformer architecture block. The procedure was later perfected by the Swin Transformers [66] which uses a shifted window to apply the self-attention mechanism, mimicking the convolution operation of CNN and also improving the performance on vision-related tasks. The task of combining the feature vectors primarily used element-wise multiplication, but recent models combine the image and textural modalities followed by the success of CLIP. We shall now delve into some of the standard VQA approaches which evolved throughout the years.

2.4.1 VQA: Visual Question Answering [1]

The most revolutionary paper in the domain of VQA introduces the task of answering free-form and open-ended questions with an image as the context and introduced a dataset of the same name. Here, a deep learning model is given an image and a general question of any form and the model must provide an accurate natural language answer. This task is open-ended, meaning that the questions and answers can take many different forms. Visual questions may target specific areas of an image, requiring the model to have a comprehensive view of the image and the ability to use complex reasoning. VQA is different from tasks like image captioning, which require a more general understanding of the image. The authors provide a dataset containing approximately 0.25 million images, 0.76 million questions, and 10 million answers for training and evaluating VQA models.

Proposed Method

In the paper, the proposed model leverages a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to process both image and text inputs. The CNN is responsible for extracting visual features from the image, while the LSTM encodes the natural language question into a fixed-length vector

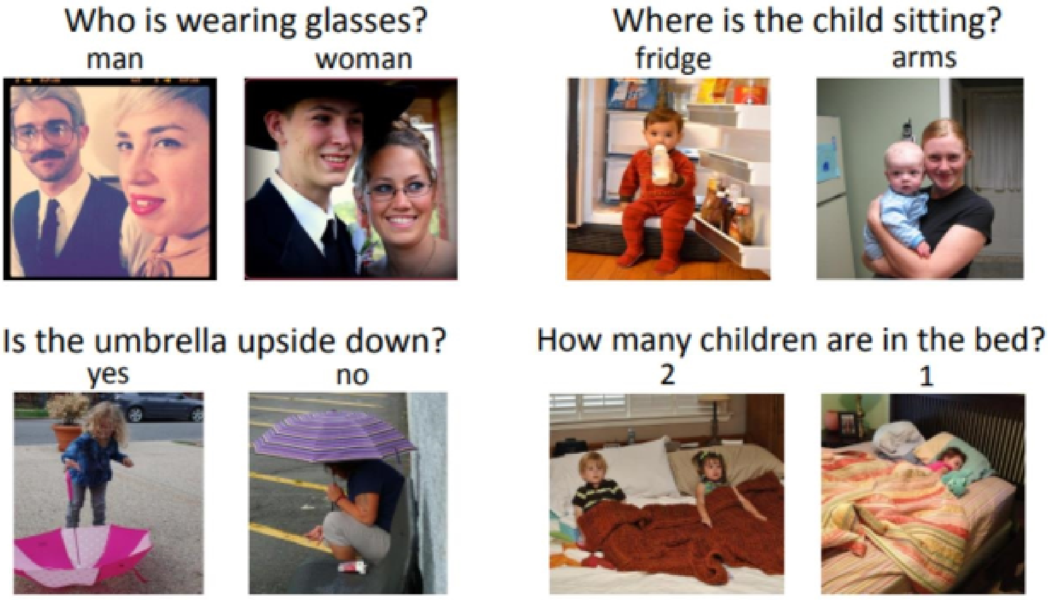

Who is wearing glasses?
man
woman
Where is the child sitting?
fridge
arms
Is the umbrella upside down?
yes
no
How many children are in the bed?
2
1

Figure 2.6: Sample questions and images from the VQAv2 dataset [15]

representation. These two representations are then merged and fed into a multi-layer perceptron (MLP) to generate the final answer. To train the model, a large dataset called VQA was utilized, which consists of a vast number of images along with corresponding questions and answers. The model was trained to minimize the cross-entropy loss between the predicted answers and the ground truth answers during training.

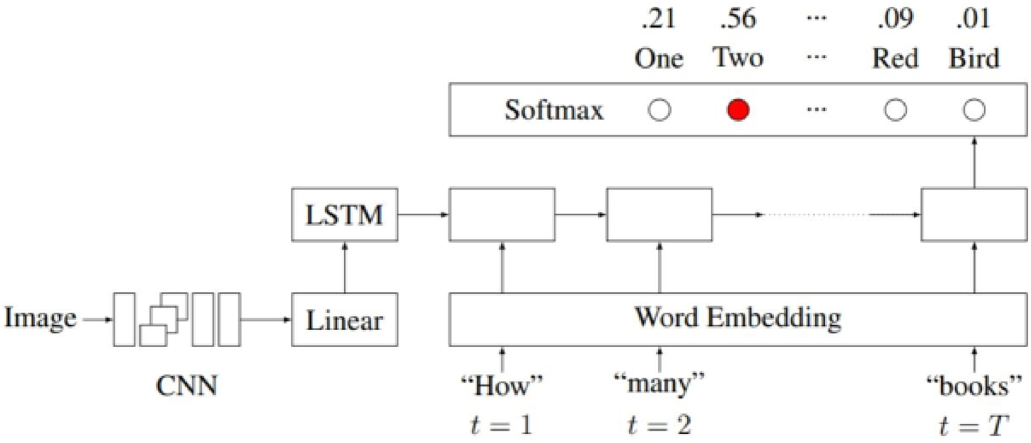
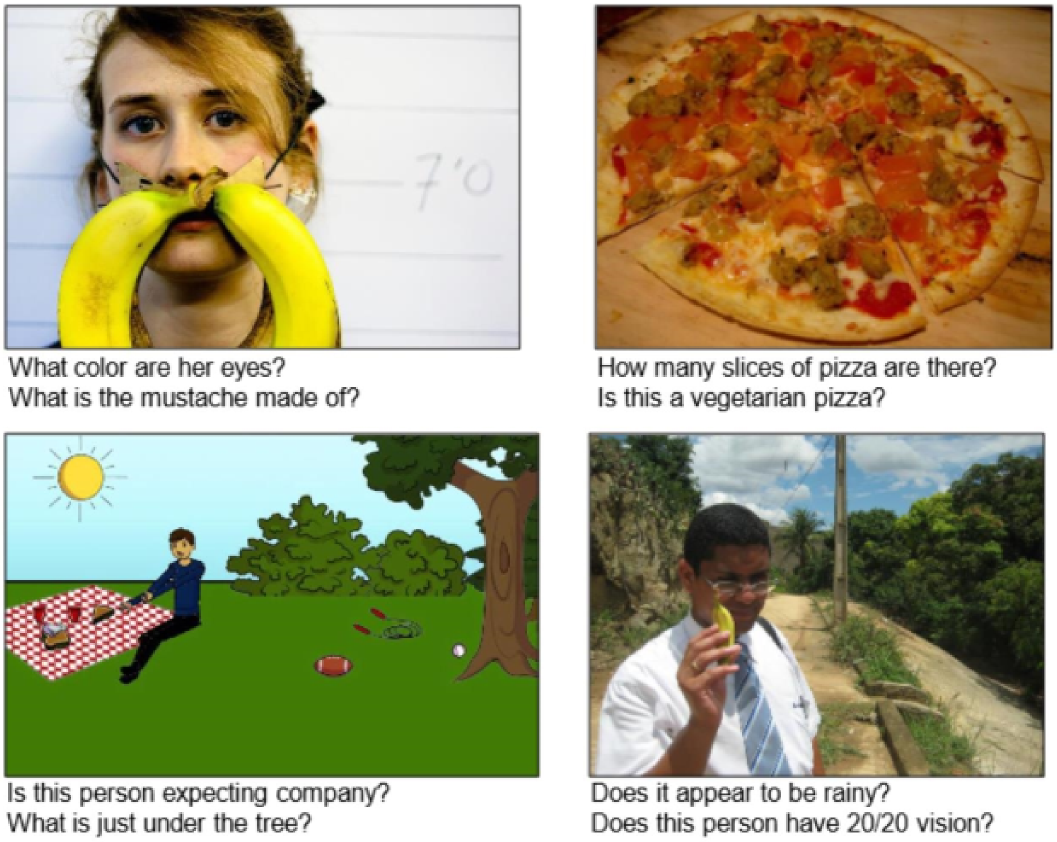


Figure 2.7: Baseline model proposed in [1] which uses CNN [16] and LSTM [17] for image and question feature extraction, and generates the answer based on a trainable softmax layer.

Dataset

The dataset is an amalgamation of real-life images with abstract scenes. The distribution of questions showed a surprising variety of types, including W-type questions i.e., “What”, “Why”, “Where”, “How”, etc., and also questions starting with “Is”, “Do”, “Does”, etc. The authors also provide several examples of questions and answers and note that "What is...” questions are particularly interesting because they have a wide variety of possible answers.



What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

Is this person expecting company?
What is just under the tree?

Does it appear to be rainy?
Does this person have 20/20 vision?

Figure 2.8: The paper by Goyal et al. [1] showcases open-ended questions collected from Amazon Mechanical Turk for Visual Question Answering (VQA). These questions require a combination of visual understanding and common-sense knowledge to provide accurate answers.

Contributions

1. The VQA paper came up with a dataset that revolutionized the field of VQA by asking open-ended questions and providing multiple human-generated answers to those open-ended questions
2. Realistic and abstract images with various question types create a perfect blend for a dataset suitable to train a generalized VQA model.

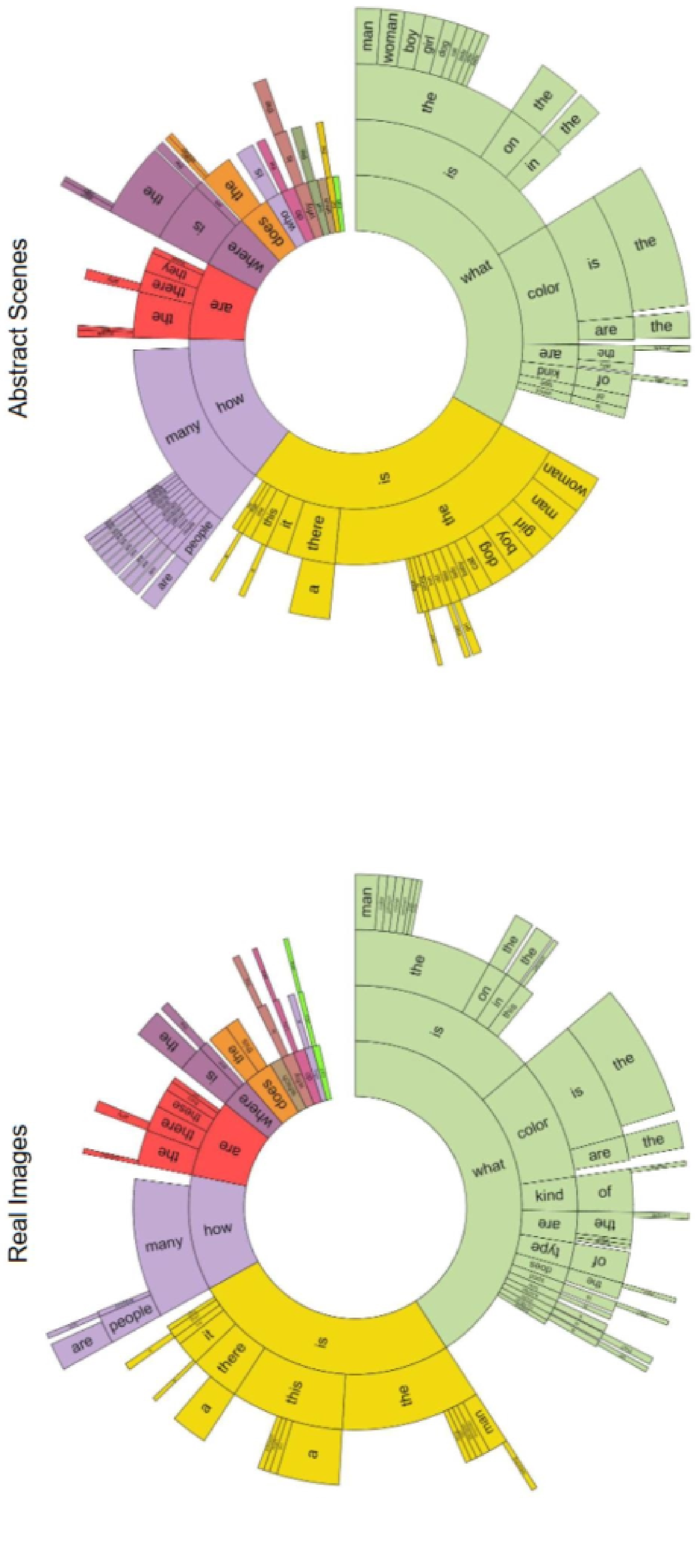


Figure 2.9: Distribution of the types of questions for real and abstract scenes in [15]

3. The CNN+LSTM-based model has been a standard deep learning-based approach, often referred to as joint embedding methodology for VQA and has been popular for many years till the advent of transformers.

Limitations

1. While the paper presents promising results, the VQA model may encounter difficulties when faced with a wide range of visual content, particularly in complex or uncommon scenarios. The model's ability to generalize to diverse and nuanced images, thereby providing accurate answers may be limited.

2. VQA models can be sensitive to biases present in the training data used to develop them. Biases inherent in large-scale VQA datasets may affect the model's performance by promoting certain answer choices over others, potentially leading to biased or skewed responses.

3. VQA models often struggle with comprehensive contextual understanding, particularly in cases where questions require deeper reasoning or implicit knowledge. The models may rely heavily on superficial cues within images or question phrasing, potentially leading to incorrect or inadequate answers.

2.4.2 Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [2]

Anderson *et. al.* [2] debates over devising a new type of attention mechanism that tries to mimic the human visual system of both top-down and bottom-up attention. In their work, they take on a very similar terminology and call attention mechanisms powered by task-specific context as *'top-down'*, and purely visual feed-forward attention mechanisms as *'bottom-up'* [2]. In the typical approach, attention mechanisms in VQA models are trained to focus on specific parts of the image or partially completed captions based on their relationship to the context. This is often achieved by selectively attending to the output of certain layers in a convolutional neural network (CNN). However, one limitation of this approach is that it overlooks the crucial aspect of determining which image regions should be the focus of attention.

The process of identifying the relevant image regions for attention deserves more consideration and exploration in order to enhance the performance of VQA models. As seen in fig-2.10, the input regions in VQA models are defined as a grid of equally sized and shaped neural receptive fields. However, this approach treats all regions of the image equally, regardless of their content or relevance. To achieve more human-like captions and question answers, it is more appropriate to focus attention on objects

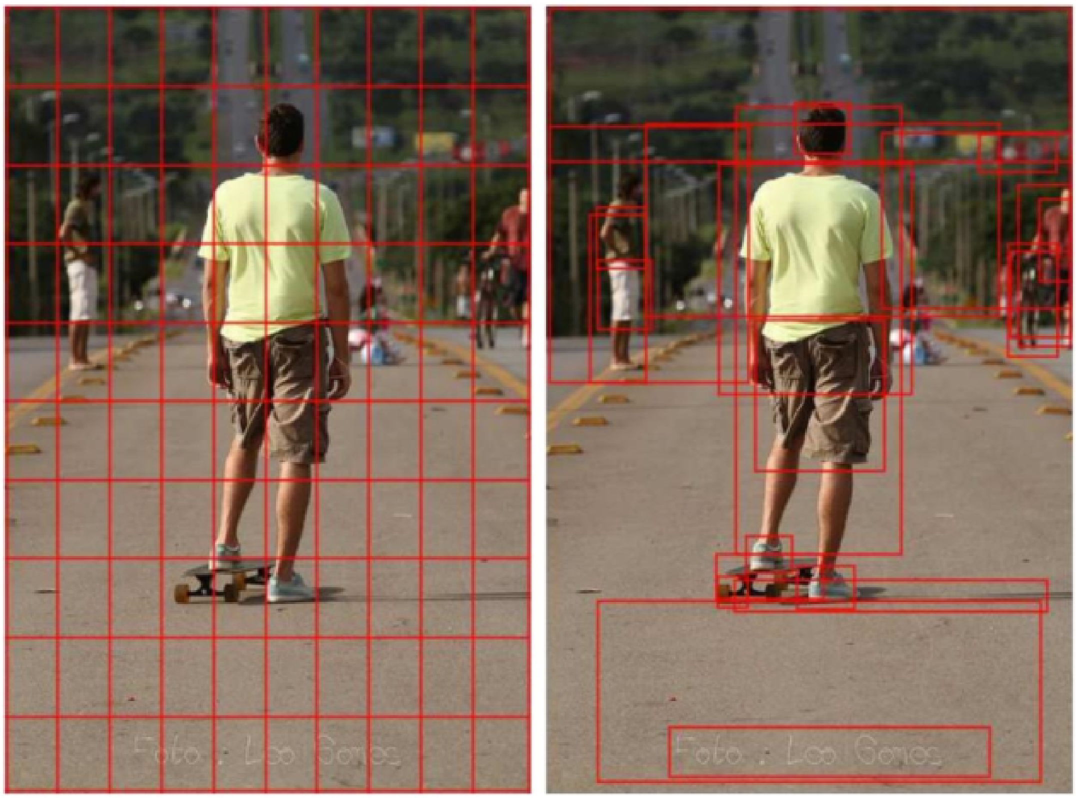


Figure 2.10: In traditional attention models used in computer vision, the focus is on equally-sized image regions represented by CNN features (left) [2]. However, the authors of this study introduced a novel approach (right) that enables attention calculations to be performed at the level of specific objects and other significant regions in the image.

and other visually salient regions within the image. By prioritizing these meaningful elements, VQA models can generate more contextually appropriate and insightful responses.

In their study, the researchers introduce a novel approach that combines both bottom-up and top-down visual attention mechanisms. The bottom-up mechanism focuses on identifying salient regions within the image by generating a set of region proposals, each represented by a pooled convolutional feature vector. This is achieved using an object detection framework like Faster R-CNN [67], which naturally embodies the concept of bottom-up attention. On the other hand, the top-down mechanism utilizes task-specific context to predict how attention should be distributed across the image regions. The resulting attended feature vector is then computed by taking a weighted average of the image features across all identified regions.

In their study, the researchers introduce the idea of using bounding boxes to define spatial regions and leverage the Faster R-CNN [67] model for implementing the bottom-up attention mechanism. Faster R-CNN, a specialized object detection model, is employed to identify and precisely localize objects of specific classes using bounding

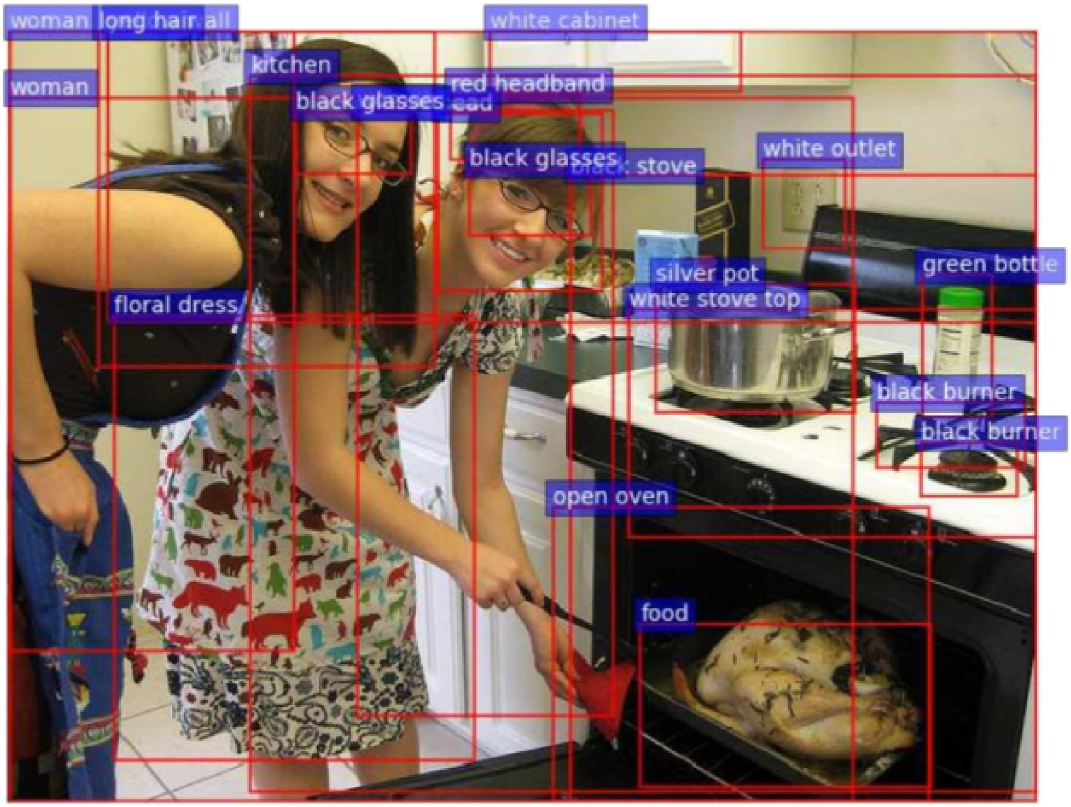


Figure 2.11: The authors of the study [2] present example output from their bottom-up attention model based on Faster R-CNN [2]. The output consists of bounding boxes that are labeled with an attribute class and an object class. It is important to note that in captioning and visual question answering (VQA), only the feature vectors are used, and not the predicted labels [2].

boxes. The object detection process in Faster R-CNN comprises two stages. Firstly, the Region Proposal Network (RPN), a small network, examines intermediate-level features from a convolutional neural network (CNN) to generate object proposals. These proposals consist of objectness scores and refined bounding box coordinates for anchor boxes of different scales and aspect ratios. The most promising box proposals are selected based on non-maximum suppression with an intersection-over-union threshold. In the second stage, region of interest (RoI) pooling is employed to extract fixed-size feature maps (e.g., 14×14) for each selected box proposal. These feature maps are then combined and processed by the final layers of the CNN. The output of the model includes a softmax distribution representing the probabilities of different class labels and class-specific refinements for each proposed bounding box.

To obtain image features suitable for tasks such as image captioning or visual question answering (VQA), a non-maximum suppression technique is applied. This involves setting an intersection-over-union (IoU) threshold to filter out redundant bounding boxes and keeping only the most relevant ones for each object class. Specifically, regions are selected if their detection probability for any class exceeds a confidence threshold. For each selected region, the mean-pooled convolutional feature is computed, resulting in image feature vectors with a dimensionality of 2048. By utilizing Faster R-CNN in this way, the model effectively functions as a "hard" attention mechanism, focusing on a small subset of image bounding box features among a large number of potential configurations.

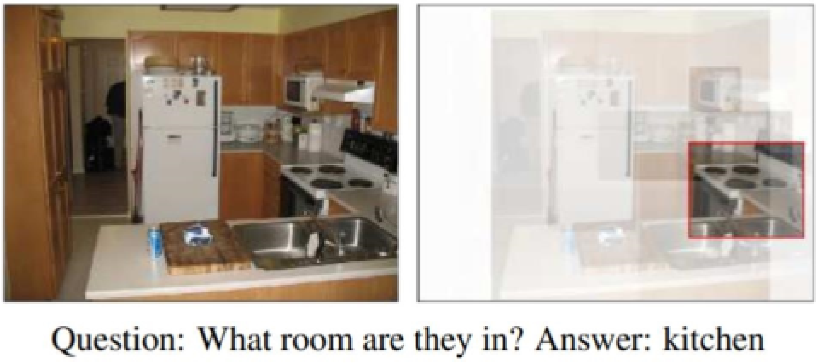


Figure 2.12: An example from the study by Anderson et al. [2] showcases the attention output in a Visual Question Answering (VQA) task. The question posed is "What room are they in?" and the model's attention is directed towards the stove-top, leading to the generated answer "kitchen" [2]. This demonstrates how the model focuses on relevant image regions to provide accurate answers in the VQA task [2].

This work presents a novel approach that bridges the gap between visual and linguistic understanding by leveraging advancements in object detection. Unlike traditional methods, which treat attention regions independently, this approach takes into account the spatial co-location of visual concepts associated with objects. By processing all the information related to an object together, a more comprehensive understanding of the visual content can be achieved. This attention mechanism holds promise for applications such as visual question answering (VQA) in a zero-shot setting, where the model needs to answer questions about images it has not been explicitly trained on. By considering all relevant visual concepts simultaneously, the model may be better equipped to handle novel scenarios and generalize its understanding across different images.

Contributions

1. The paper introduces novel attention mechanisms for image captioning and visual question-answering tasks. These mechanisms, referred to as "bottom-up" and "top-down" attention, improve the model's ability to focus on relevant image regions and generate accurate captions or answers.

2. The authors propose an approach that combines the strengths of both bottom-up and top-down attention mechanisms. This integration allows the model to effectively incorporate contextual information from the image, leading to more contextually grounded captions and answers.

3. Experimental evaluations demonstrate that the introduced attention mechanisms significantly enhance the performance of image captioning and visual question-answering models. The improved models achieve state-of-the-art results on benchmark datasets, showcasing the effectiveness of the proposed approach.

4. The bottom-up and top-down attention mechanisms provide interpretability and explainability in the image captioning and visual question-answering processes. By explicitly highlighting relevant image regions, the models offer insights into the decision-making process, enhancing the transparency of the generated captions and answers.

5. The proposed attention mechanisms show versatility and generalization across different tasks, including image captioning and visual question answering. This versatility allows the models to adapt to various vision and language tasks, showcasing their potential for broader applications beyond the specific tasks considered in the paper.

Limitations

1. The paper acknowledges that the effectiveness of the bottom-up and top-down attention mechanisms can vary depending on the specific image captioning and visual question-answering tasks considered. Factors such as dataset characteristics, image complexity, or question types may impact the overall performance and generalization capabilities.

2. The proposed attention mechanisms may be sensitive to variations in image quality, such as low-resolution or noisy images. Additionally, the models may encounter challenges when dealing with complex or ambiguous visual content, leading to potential inaccuracies or errors in generating captions or answers.

3. The performance of the bottom-up and top-down attention mechanisms heavily relies on the availability and quality of the training data. Limited or biased training data may restrict the model's ability to capture diverse visual patterns and language semantics, affecting its performance on unseen examples.

4. The introduced attention mechanisms may impose increased computational demands due to the need to process and attend to multiple image regions or incorporate contextual information. This can limit their practical applicability, particularly in resource-constrained environments or real-time applications.

5. The paper may not extensively evaluate the performance of the attention mechanisms on specialized or domain-specific datasets. This limitation restricts insights into their adaptability and effectiveness in scenarios where the image content or question types deviate significantly from the datasets used for evaluation.

2.4.3 BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation [3]

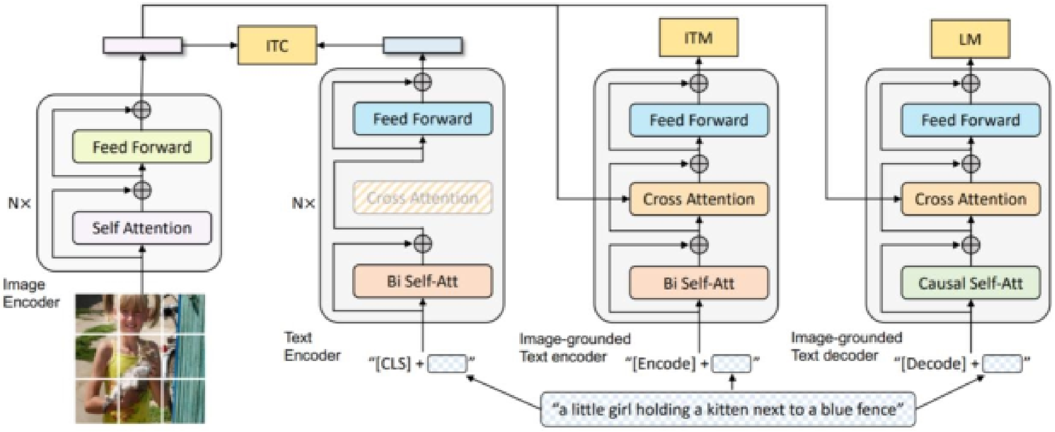


Figure 2.13: Overview of the architecture of BLIP [3]

BLIP (Bridging Language, Image, and Perception) presents a novel approach to Vision-Language pre-training that offers flexibility in both understanding and generating vision-language tasks [3]. In order to effectively leverage noisy web data, BLIP utilizes a captioning model to generate synthetic captions and employs a filtering mechanism to eliminate noisy ones. For image encoding, they adopt a visual transformer that divides the input image into patches and encodes them as sequential embeddings [65]. An additional [CLS] token is introduced in BLIP to represent the global image feature [3]. Unlike traditional methods that rely on pre-trained object detectors, BLIP opts for a Vision Transformer (ViT), which provides computational efficiency and has gained popularity in recent approaches [4].

To enable pre-training of a unified model with combined understanding and generation capabilities, the researchers proposed the Multimodal Mixture of Encoder-Decoder (MED) framework [3]. MED encompasses three key functionalities:

- Unimodal Encoder: Separate picture and text encoding is required for this feature. The text encoder used is comparable to BERT [68], which appends a [CLS] token to the beginning of the text input to provide a summary of the sentence.
- Image-Grounded Text Encoder: For each transformer block in the text encoder, a new cross-attention (CA) layer is added between the self-attention (SA) layer and the feed-forward network (FFN) in order to integrate visual information. The text is supplemented with a task-specific [Encode] token, and the ensuing embedding of [Encode] functions as the multimodal representation of the image-text pair.
- Image-Grounded Text Decoder: The image-grounded text encoder's bidirectional self-attention layers are swapped out for causal self-attention layers by this functionality. The start of a sequence is denoted by a [Decode] token, and the end of a sequence is denoted by an end-of-sequence token.

The model can work in a variety of modes thanks to these three MED features, each of which is designed for a particular job related to unimodal encoding, image-grounded text encoding, or image-grounded text decoding. They focus on three distinct loss functions: Language Modeling Loss (LM), Image-Text Matching Loss (ITM), and Image-Text Contrastive Loss (ITC).

Contributions

1. The authors propose a bootstrapping technique that enables joint learning of visual and textual representations. This approach facilitates the integration of visual and linguistic cues, enhancing the model's understanding of the relationships between images and corresponding textual descriptions.
2. BLIP demonstrates remarkable transfer learning capabilities across vision and language tasks. By leveraging pre-trained models and fine-tuning task-specific data, the proposed method achieves state-of-the-art performance on various benchmarks, highlighting the effectiveness of the approach.
3. The paper showcases the effectiveness of BLIP in generating high-quality visual and textual content. The integrated pre-training enables the model to generate accurate and coherent image descriptions, captions, and other vision-language outputs, enhancing the overall quality of generated content.
4. The contributions of the paper have implications for a wide range of applications, including image captioning, visual question answering, and visual storytelling.

The BLIP approach offers a unified framework for vision-language tasks, enabling more comprehensive and versatile understanding and generation capabilities.

Limitations

1. While BLIP demonstrates impressive performance on various benchmarks, the generalization to unseen or domain-specific data remains a challenge. The model's ability to comprehend and generate vision-language content may vary when encountering novel or specialized examples not adequately represented in the training data.

2. The effectiveness of BLIP is influenced by the biases present in the training data. Biases may affect the model's understanding and generation capabilities, potentially leading to skewed or undesired outputs. Addressing dataset biases is crucial to ensure fair and unbiased vision-language understanding and generation.

3. The BLIP method involves significant computational requirements due to the joint learning of visual and textual representations. Training and fine-tuning large-scale models can be computationally expensive and may limit the practical applicability of the approach, particularly in resource-constrained environments.

4. While BLIP achieves impressive results, the inner workings and interpretability of the model's decision-making process remain challenging. Understanding how the model combines visual and textual information to generate outputs is crucial for transparency and trustworthiness, but further research is needed to enhance interpretability.

5. The paper may not extensively evaluate the performance of BLIP on specialized domains or narrow tasks. Understanding the model's behavior and performance in such scenarios is important to ensure its effectiveness across diverse application domains.

2.4.4 ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision [4]

ViLT is a compact Vision-and-Language Transformer (ViLT) model, which can be considered monolithic as it significantly simplifies the processing of visual inputs. The authors adopt a convolution-free approach for visual input processing, aligning it with the manner in which we handle textual inputs. ViLT exhibits competitive or even superior performance on downstream tasks when compared to existing models.

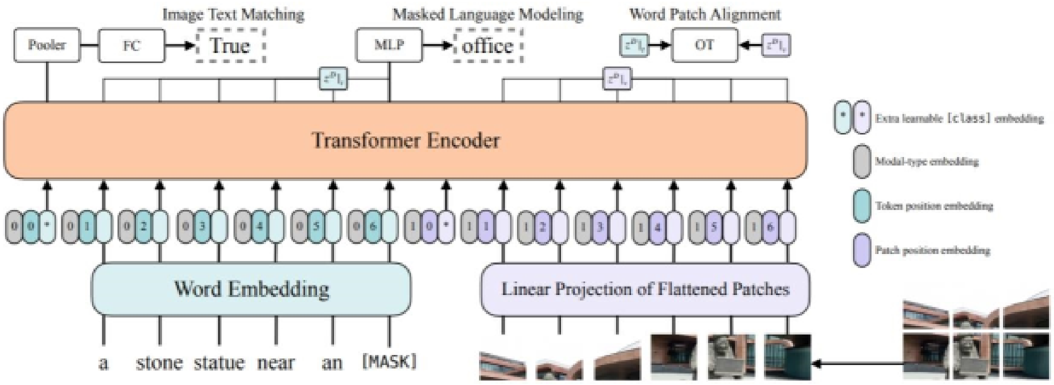


Figure 2.14: Architecture of ViLT [4]

ViLT features a concise architecture as a Vision-and-Language Pretraining (VLP) model, incorporating a streamlined visual embedding pipeline and adopting the single-stream approach. Unlike previous approaches in the literature, it deviates by initializing the weights of the interaction transformer from a pre-trained Vision Transformer (ViT) model [65], rather than from BERT [68]. This initialization strategy leverages the strength of the interaction layers in effectively processing visual features, eliminating the need for a separate deep visual embedder. This approach allows for efficient integration of visual and textual information within a unified framework. It is pre-trained on two tasks: Image-text matching and Masked language modeling.

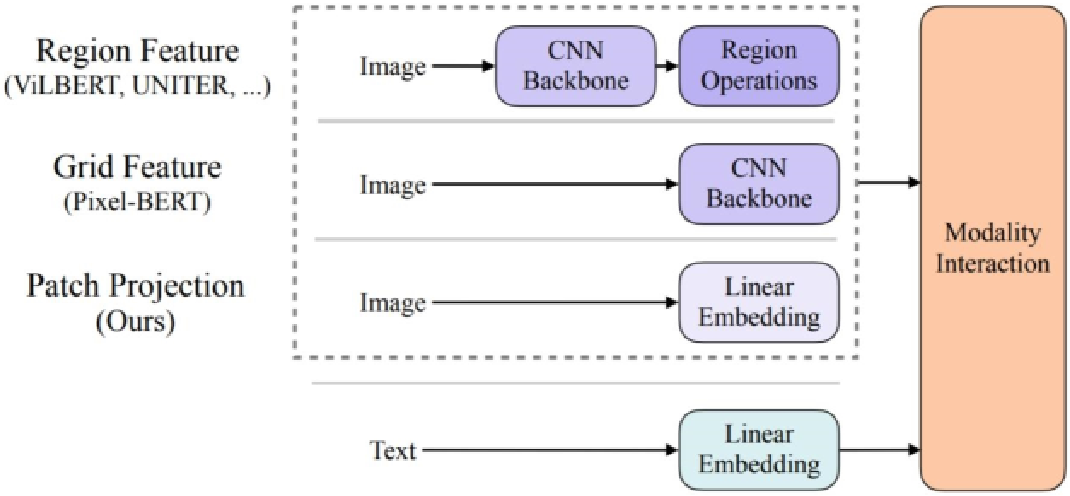


Figure 2.15: Overview of Visual Embedding Schema in ViLT [4]

Contributions

1. The paper introduces an innovative method, called ViLT, for vision-and-language understanding without relying on convolutional neural networks (CNNs) or region supervision. This approach explores new avenues for integrating vision and language modalities using a transformer-based architecture.

2. ViLT offers a unified framework that seamlessly integrates visual and textual information using transformers. This end-to-end integration enhances the model's ability to capture complex cross-modal dependencies, enabling more comprehensive vision-and-language understanding.

3. The authors propose a weakly supervised learning approach that eliminates the need for explicit region-level annotations or reliance on pre-trained CNNs. By leveraging large-scale image-text datasets, ViLT learns to align and associate visual and textual features without explicit pixel-level or region-level supervision.

4. Experimental evaluations demonstrate that ViLT achieves competitive performance on a range of vision-language tasks, including image-text retrieval, image captioning, and visual question answering. The results highlight the effectiveness of the proposed approach in capturing the rich interactions between vision and language modalities.

5. The contributions of the paper have implications for various applications requiring vision-language understanding, such as image retrieval, multimedia analysis, and content recommendation. ViLT's transformer-based architecture provides a flexible and versatile framework that can be adapted to different vision-language tasks and domains.

Limitations

1. The ViLT framework involves computationally intensive transformer-based architectures, which can be demanding in terms of memory and computational resources. The training and inference processes may require significant computational infrastructure, limiting the practical deployment of ViLT in resource-constrained settings.

2. While ViLT achieves competitive results, understanding the decision-making process and internal representations of the model remains challenging. Interpreting how the model integrates vision and language information and the reasons

behind its predictions is crucial for transparency and trust, requiring further research in interpretability techniques.

3. The paper may not extensively evaluate ViLT's performance on specific domains or narrow vision-language tasks. Assessing the model's behavior and performance in specialized scenarios is important to understand its strengths and limitations in domain-specific applications.

2.4.5 GIT: A Generative Image-to-text Transformer for Vision and Language [5]

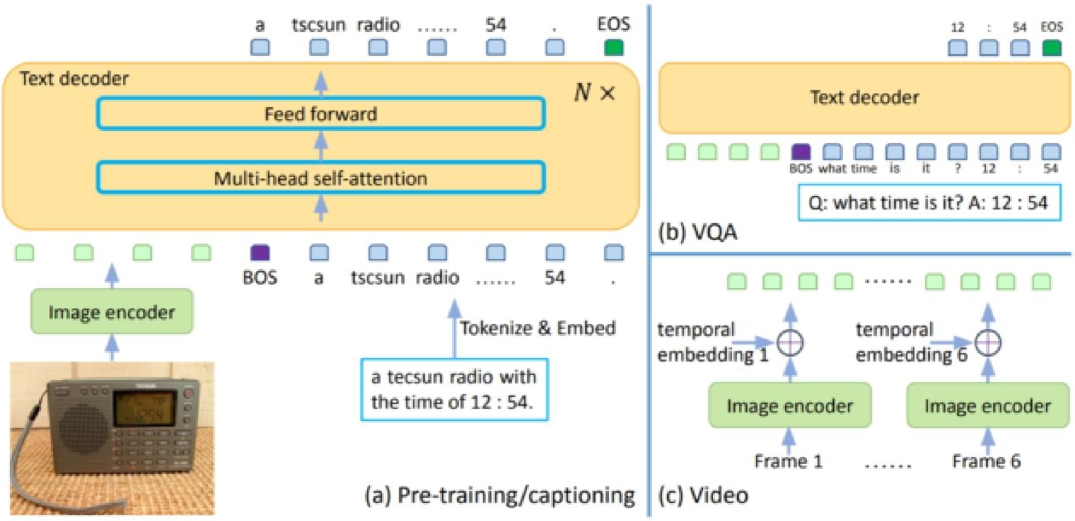

a tscsun radio ...... 54 . EOS
Text decoder
N ×
Feed forward
Multi-head self-attention
BOS a tscsun radio ...... 54 .
Image encoder
Tokenize & Embed
a tecsun radio with the time of 12 : 54.
(a) Pre-training/captioning
12 : 54 EOS
Text decoder
BOS what time is it ? 12 : 54
Q: what time is it? A: 12 : 54
(b) VQA
temporal embedding 1
temporal embedding 6
Image encoder
Image encoder
Frame 1 ...... Frame 6
(c) Video

Figure 2.16: Architecture of GIT [5]

GIT (Generating Informative Text) is a Transformer model introduced by Wang et al. [5] that incorporates visual inputs alongside text by leveraging the vision encoder from CLIP [9]. The authors simplify the architecture by using a single image encoder, a single text encoder, and a single pre-training task, namely masked language modeling. When applied to Visual Question Answering (VQA), the input question is treated as a text prefix, and the model generates an auto-regressive answer.

During the fine-tuning phase, the query and the ground truth answer are combined and treated as a specific caption within the context of the visual question response. To train the model to generate accurate responses, the language modeling (LM) loss is specifically applied to the answer tokens and the [EOS] (end-of-sequence) token. During inference, the question is used as the caption prefix, and the model generates the remaining information to complete the response prediction. This approach enables the model to generate answers based on the provided visual context. Unlike existing techniques that rely on pre-defined candidate solutions and frame the problem as a classification task, GIT employs generative operations. This means that even during inference, the model does not rely on pre-collected response candidates and is capable of producing multiple tokens, including the solution and the [EOS] token, as part of the answer prediction [5].

Contributions

1. The paper introduces a novel generative model, referred to as GIT, for transforming images into textual descriptions. The GIT model leverages an innovative architecture based on transformers to capture the complex interactions between visual and language modalities, resulting in the generation of coherent and contextually grounded image captions.

2. GIT successfully integrates visual information from images with linguistic cues, enabling the model to generate accurate and meaningful text descriptions. By effectively combining these modalities, GIT learns to extract rich representations and produce high-quality image captions that capture the essence of the visual content.

3. The proposed GIT model adopts an advanced transformer-based architecture, which facilitates the modeling of long-range dependencies and the relationships between different elements within images and texts. This architecture enables the generation of more coherent, fluent, and contextually relevant image descriptions.

Limitations

1. GIT's performance can be influenced by biases present in the training data used to develop the model. Biases inherent in large-scale image-text datasets may lead to biased associations between visual and textual features, potentially affecting the accuracy and diversity of the generated image captions.

2. The GIT model's transformer-based architecture can demand significant computational resources during both the training and inference phases. The computational complexity associated with these models may limit their practical deployment, particularly in resource-constrained environments or real-time applications.

3. While GIT achieves impressive results, understanding the decision-making process and inner workings of the model remains challenging. Interpreting how the model integrates visual and language information to generate image captions

requires further research to enhance transparency, interpretability, and trustworthiness.

2.5 Visual and Textual Robustness

2.5.1 A Novel Framework for Robustness Analysis of Visual QA Models [6]

Ideally, this paper is closely similar to the research problem we are trying to solve. To apply VQA models in a realistic environment, the input can be affected by noise. The noise can affect both the input question and the input image. This paper deals with the noise associated with the input questions and also comes up with a metric they introduced in another paper called the R-score. The noise to questions can be thought of in multiple ways – for instance, paraphrasing the question can be a form of noise that shouldn't change the expected output answer from the model. For example – let the original question, "What is the color of the banana?" can be paraphrased as "Which color is associated with the banana in the given image?" Both questions have the same answer, "Yellow", and have the same meaning. But, a VQA model might associate some words from the input question with some form of an answer. This problem is similar to the overfitting of neural networks or any simple machine learning model. A generalized and robust VQA model should be able to correctly answer such paraphrased questions.

Proposed Framework

The type of noise discussed in this paper is not similar to paraphrasing, instead, the original input question is compared to a set of questions called the basic questions from a dataset called Basic Questions Dataset (BQD). A ranking of the input question with the basic questions is performed based on the similarity and the top ones, e.g., the top 3 are taken. Afterward, the Basic questions are appended to the input questions to form the noisy questions. To measure the accuracy, they also provide a new metric called the R-Score. R-Score can be defined as:

$$R_{score} = clamp_0^1 \frac{\sqrt{m} - \sqrt{Acc_{diff}}}{\sqrt{m} - \sqrt{t}} \tag{2.1}$$

Here, $clamp_b^a(x) = max(a, min(b, x))$ where, $0 \leq t < m \leq 1$. Here, t is the tolerance and m is the maximum robustness limit.

Being a general framework, the noise type can be changed to other forms of noise, e.g., word-shuffling in questions, grammatical errors, and so on. These forms of noise can be controlled by the Hamming distance, and thus several levels of noise can be

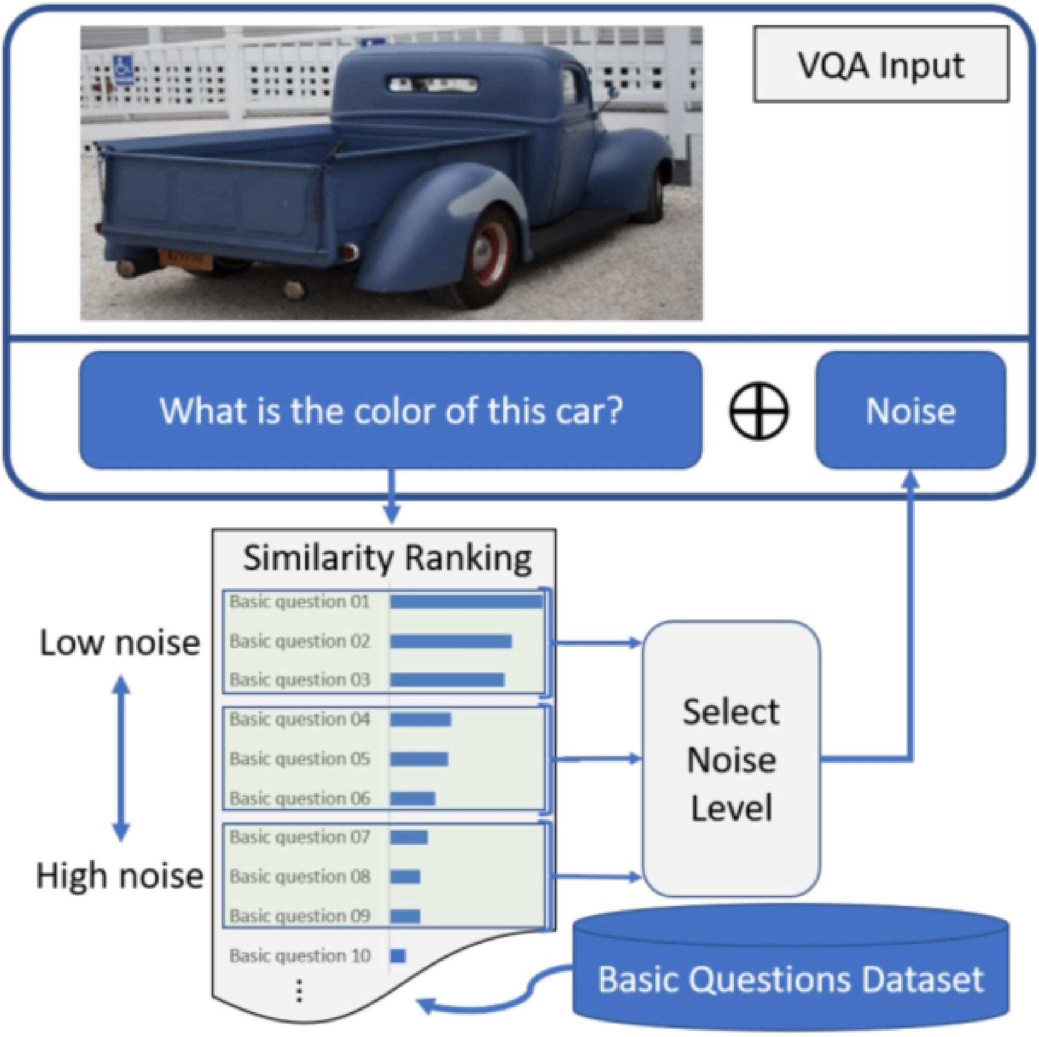


Figure 2.17: Framework proposed by [6] to test the robustness of textural inputs of the VQA models

proposed. On the other hand, question paraphrasing noise isn't controllable and can be called trivially controllable noise. Hence, multiple levels of paraphrasing noise are not possible. To make a properly robust model, the authors propose to incorporate different types of noise at various levels.

Contributions

1. The paper proposes a novel framework that is modular and has tested its framework on six different models.
2. Proposed a text-based method to rank similarity which is comparable to derivations of BLEU
3. Proposed a new sentence evaluation metric called the R-score.

Limitations

1. This work only discusses methods for evaluating the lingual robustness of various VQA methods.

2. The authors evaluate their method on dated VQA methods. No relevant modern models were used.

3. They provide only one metric as the basis for determining the robustness of a model.

2.5.2 Benchmarking Neural Network Robustness to Common Corruptions and Perturbations [7]

ImageNet-C where C stands for Corruption is an augmented version of the famous ImageNet dataset. The image augmentation is performed to test the visual robustness of standard computer vision models and is particularly targeted towards the Neural Network based models from AlexNet to ResNets. Their work has been motivated by the robustness of the human vision system, which is not fooled by minute changes in an image and is correctly able to classify an image at high levels of distortion, poor lighting conditions, and other visual challenges. However, neural networks have been particularly prone to such obstacles. A famous example is the addition of 9% Nematode noise to an image of a panda which would, still, visually look like a panda. Surprisingly, the Convolutional Neural Networks (CNNs) confidently classified that as some other object, something that humans would never do. These forms of adversarial noise are primarily designed to trick neural networks into misclassification as neural networks can not naturally "see" but has to rely on the pixel values to update their weights. However, creating misleading patterns within the pixel values can often result in the neural network confidently misclassifying an object, and such a problem can be deadly in real-life scenarios where the neural networks are assigned to more responsible tasks.

Dataset

The ImageNet-C primarily contains 15 types of noise which are algorithmically generated forms of corruption. If we start from the beginning we will see standard forms of noise in digital image processing such as Gaussian noise which adds a random value following the Gaussian distribution to the current pixel values, and impulse noise which randomly changes a pixel value to a solid-colored pixel, and so on. Again, there are various forms of blurring noise like defocus, frosted glass blur, and motion blur

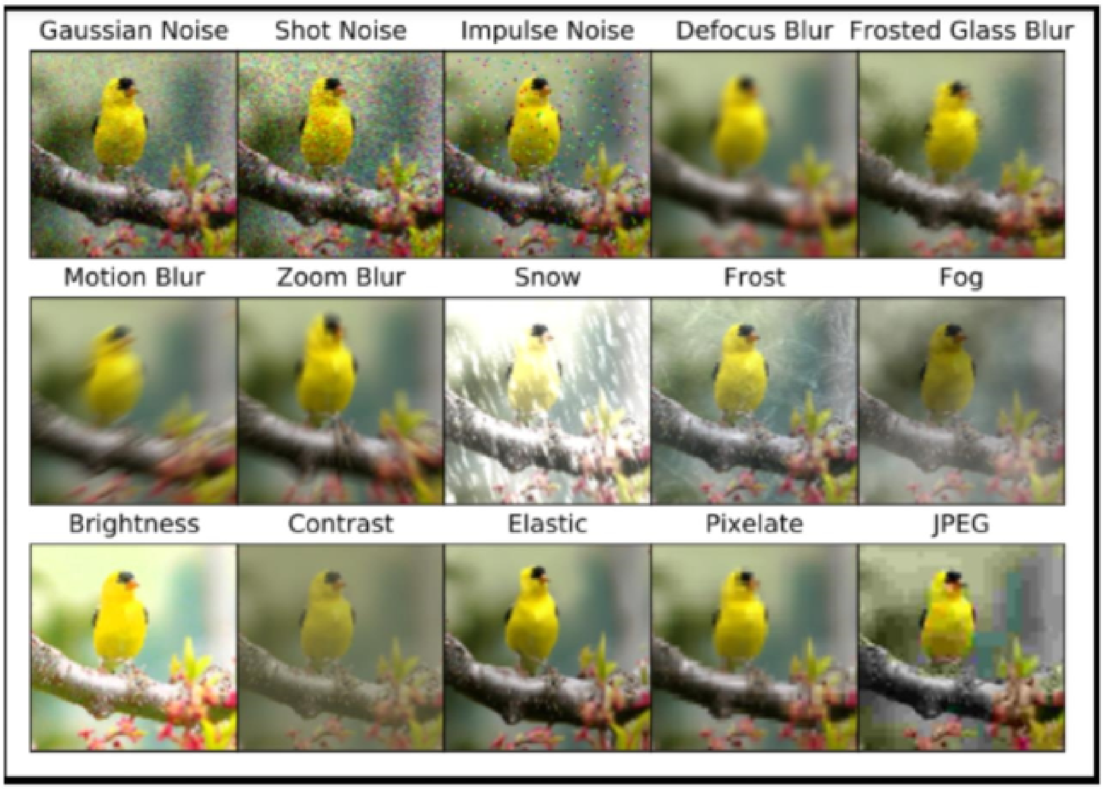


Figure 2.18: The effects of the image transformations proposed in [7]

– all are pretty much self-explanatory. Forms of weather effects like snow, frost, and fog are also added to replicate realistic image conditions. Image processing effects like changing the brightness and contrast of the image are also added and finally, lossy compression effects like pixelating and JPEG compression are included. Every form of noise has 5 severity levels resulting in 75 distinct corruption types. 75 versions of ImageNet can be produced, and for each dataset, the accuracy can be measured for a specific model.

If we look at the reasoning behind the common corruption types, we can find how they are related to realistic scenarios. The most common corruption type is the Gaussian noise which appears in low lighting conditions. The Poisson noise which is also referred to as the Poisson noise occurs due to the discretization of light. We might have been familiar with the salt and pepper noise which changes a pixel value randomly to complete white or complete black. The color analog of the salt pepper noise is referred to as the Impulse noise and is a common corruption caused by bit errors. Defocus blur occurs when the object is out of focus in the image and frosted glass blur occurs due to looking at the object from a window panel covered by frosted glass. The other forms of noise have similar reasoning, and we shall not discuss in detail the realistic background associated with these noise types.

Apart from added noise levels perturbations have also been added. The concept of perturbation is a bit different from that of corruption. Perturbations are continuous incremental changes to the image, such changes can be in the form of slight tilts, slight changes in brightness, and so on. A model's robustness on perturbation is also a valid measure in testing visual robustness. However, perturbations are computationally more expensive, and performing many subtle changes on our large VQA dataset would be extremely challenging and resource intensive. Perturbations are also associated with the robustness of videos more than images and as we are not dealing with Video Question Answering, we shall skip the details of implementing perturbations to our datasets.

Contributions:

1. Designing a framework for testing visual robustness for varying noise types at varying noise levels
2. Designing another framework for testing visual robustness through perturbations that create incremental subtle changes to the images
3. Providing access to the framework, and the associated transformation functions in a GitHub code repository.

Limitations:

1. The authors evaluate their method on dated VQA methods as the availability of modern transformers-based models was not prevalent at that time.
2. They provide only one metric as the basis for determining the robustness of a model.

2.5.3 CARETS: A Consistency And Robustness Evaluative Test Suite for VQA [8]

CARETS is a systematic test suite designed to assess the consistency and robustness of contemporary VQA methods. It has six thorough capacity tests, each of which focuses on a different skill, such as rephrasing, logical symmetry, or picture obfuscation. CARETS uses a balanced question-generating approach to generate pairs of examples for model evaluation, in contrast to conventional VQA test sets. This study is the most comparable to our proposed work from a similarity perspective because they also present a framework that creates tests for fine-grained capability testing on the textual modality. The authors describe their methodology as a systematic test suite that uses a set of six fine-grained capability tests to evaluate the consistency and dependability of contemporary VQA (Visual Question Answering) models. CARETS, in contrast to current VQA test sets, uses balanced question generation to produce pairs of examples to test models, with each pair concentrating on a different capability such as rephrasing, logical symmetry, or picture obfuscation [8]. The authors assess six contemporary VQA systems using CARETS, and they find a number of remediable flaws in model understanding, particularly when it comes to ideas like negation, disjunction, or hypernym invariance. In CARETS, each test point consists of two occurrences that are minor but purposeful changes of one another, either aesthetically or in the wording of the question. This makes it possible to evaluate capabilities at a finer scale than only by looking at high accuracy ratings.

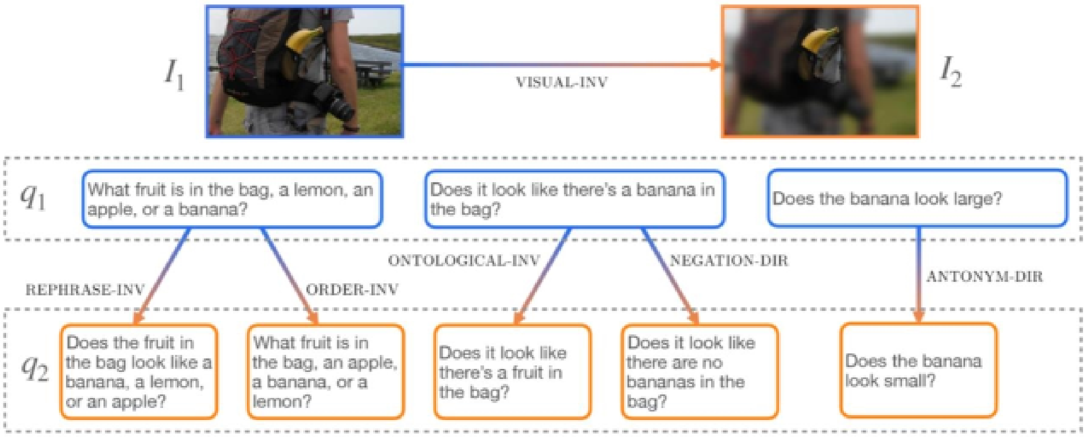

I_1
VISUAL-INV
I_2
q_1
What fruit is in the bag, a lemon, an apple, or a banana?
Does it look like there's a banana in the bag?
Does the banana look large?
REPHRASE-INV
ORDER-INV
ONTOLOGICAL-INV
NEGATION-DIR
ANTONYM-DIR
q_2
Does the fruit in the bag look like a banana, a lemon, or an apple?
What fruit is in the bag, an apple, a banana, or a lemon?
Does it look like there's a fruit in the bag?
Does it look like there are no bananas in the bag?
Does the banana look small?

Figure 2.19: The consistency and robustness test suite (CARETS) proposed in [8].

Proposed Framework

Their first set of tests comprises four invariance tests. These tests involve modifying the phrasing of the questions while expecting the model to generate the same answer for both questions within a pair of instances.

1. **Rephrasing invariance:** The Rephrasing Invariance test evaluates the model's comprehension of minor textual modifications that preserve the meaning. For instance, it assesses the model's ability to understand and answer questions such as "What color is the bottle on the shelf, white or blue?" and "Does the color of the bottle on the shelf appear more white or blue?" Both questions convey the same meaning but differ slightly in their phrasing.

2. **Ontological invariance:** The Ontological Invariance test evaluates the model's understanding of ontology, specifically assessing its ability to recognize changes between hyponyms and hypernyms. For example, it tests whether the model can

handle a modification from "Do you see a green jacket?" to "Do you see any green clothes?" The test aims to determine if the model can accurately interpret the broader concept (hypernym) and still provide a correct answer.

3. **Order invariance:** The Order Invariance test assesses the model's understanding of logically equivalent questions that feature different orders of arguments. For example, it evaluates whether the model can comprehend and provide consistent answers to questions such as "Is the black vehicle a van or a truck?" and "Is the black vehicle a truck or a van?" Despite the variation in the order of the options, the questions convey the same logical meaning. The test aims to determine if the model can maintain consistency in its responses regardless of the argument order.

4. **Visual obfuscation invariance:** The Visual Obfuscation Invariance (VOI) test is designed to assess the model's ability to answer questions even when parts of the image that are not directly relevant to the visual question are obscured or removed. This evaluation involves applying techniques such as blurring, masking, and cropping to modify the image. By examining the model's performance in answering questions based on visually obfuscated images, the test aims to evaluate the model's capability to concentrate on the pertinent visual information and provide accurate answers despite any visual distractions or alterations that may be present.

They also created directional expectation tests to measure model behavior on instance pairs where the answer is expected to change.

1. **Attribute antonym directional expectation:** The Attribute Antonym Directional Expectation test assesses the model's comprehension of antonyms. Specifically, it evaluates the model's understanding of how changing an attribute's antonym affects the question. For instance, the test involves questions such as "Do you think that the wood table is short?" and "Do you think that the wood table is long?" These questions examine whether the model can correctly interpret the opposite meaning of the attribute and provide appropriate answers accordingly. The test aims to evaluate the model's understanding of antonyms and its ability to adjust its responses based on the directional change of attributes.

2. **Negation directional expectation:** The Negation Directional Expectation test evaluates a model's understanding of negation. It assesses the model's ability to comprehend the impact of negation on the meaning of a question. For example, the test includes questions like "Are there any apples in this picture?" and "Are

there no apples in this picture?" These questions explore whether the model can accurately interpret the presence or absence of apples based on the negation used in the question. The test aims to determine the model's grasp of negation and its ability to provide appropriate responses considering the negated context.

Contributions:

1. Designed a framework for testing the consistency and robustness of contemporary VQA methods.
2. Generated balanced dataset.
3. Introduced novel metrics for consistency and robustness.
4. Access to the framework and transformation functions have been provided in a GitHub code repository [8]

Limitations:

1. This work focuses mostly on lingual robustness and consistency testing.
2. The invariance test for testing visual consistency is not a thorough exploration and is lacking further experiments.
3. Runs experiments only on dated models.
4. Performs tests on only the GQA dataset [69].

2.6 Zero-Shot VQA (ZS-VQA)

Recent years have witnessed unprecedented performance gains on many natural language reasoning tasks, especially in zero-shot and few-shot settings, being derived from scaling up pre-trained language models (PLMs) and their training data [59, 68, 70–73]. As these models are trained on vast amounts of data, often encompassing trillions of training examples scraped from varying sources all over the internet. But before we explore modern methods that are used for zero-shot VQA, we need to examine the beginnings of VQA in general.

2.6.1 Learning Transferable Visual Models From Natural Language Supervision [9]

Radford *et. al.* [9] presented a method of connecting images and their captions found commonly on the internet. They named this method CLIP. The main goal of this work

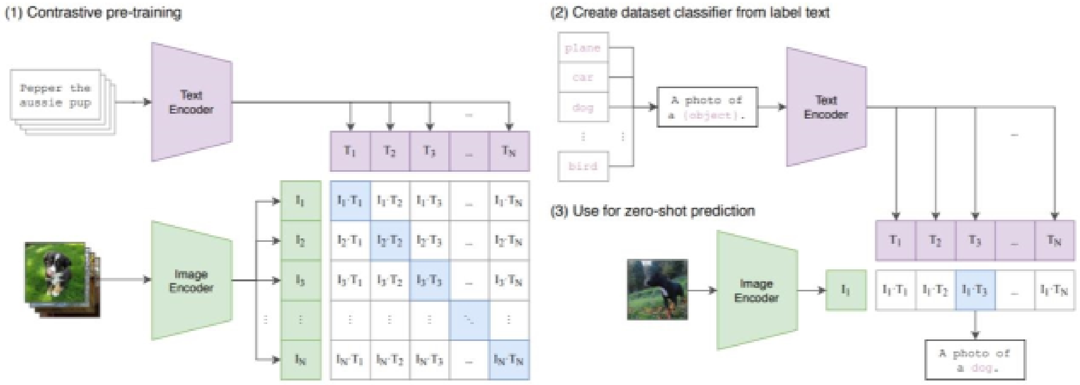

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 … T_N
I_1 I_2 I_3 ⋮ I_N
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ … $I_1{\cdot}T_N$
$I_2{\cdot}T_1$ $I_2{\cdot}T_2$ $I_2{\cdot}T_3$ … $I_2{\cdot}T_N$
$I_3{\cdot}T_1$ $I_3{\cdot}T_2$ $I_3{\cdot}T_3$ … $I_3{\cdot}T_N$
$I_N{\cdot}T_1$ $I_N{\cdot}T_2$ $I_N{\cdot}T_3$ … $I_N{\cdot}T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
T_1 T_2 T_3 … T_N
I_1
$I_1{\cdot}T_1$ $I_1{\cdot}T_2$ $I_1{\cdot}T_3$ … $I_1{\cdot}T_N$
A photo of a dog.

Figure 2.20: The CLIP method, described in the paper by Radford et al. [9], introduces a novel approach for training image and text encoders in a joint manner. Unlike standard image models that focus on predicting specific labels, CLIP trains an image encoder and a text encoder together to predict correct pairings of (image, text) examples in a batch during training. This enables the model to learn a powerful representation of both images and text. During testing, the learned text encoder is used to synthesize a zero-shot linear classifier by embedding the names or descriptions of classes from the target dataset, allowing for accurate classification without the need for fine-tuning. This approach expands the capabilities of the model to understand and relate images and text in a more versatile manner.

was to show that multitudes of downstream tasks can be transferred through a zero-shot setting where natural language is used as supervision. They use the contrastive loss to align the image and text embedding spaces so that the images represent their captions in the embedding space. It is different from regular image processing methods as it does not jointly train an encoder along with a linear layer. Rather, it trains an image encoder and a text encoder simultaneously. This training method is paired with the contrastive loss so that it can predict the correct pairings of images and captions.

CLIP calculates image and text embeddings using ViT [65] and BERT [68]. These produce embeddings in different embedding spaces and do not lead to the same vector. So there is a disconnect between the two modalities. CLIP, through contrastive loss, aims to move these vectors closer together. This is achieved by maximizing the cosine similarity of an image and its caption and minimizing the cosine similarity of the same image and every other caption. After training, the vision encoder and the textual encoder can be used together or separately in many downstream tasks often in zero-shot settings. The authors, inspired by the zero-shot capabilities of GPT-3 [59], wanted to perform a great variety of tasks while not explicitly optimizing for their benchmarks.

After training on 400 million image-caption pairs, CLIP gained excellent zero-shot transfer capabilities. They shift their focus from representation-learning capabilities of zero-shot systems to rather learn to generalize to different tasks and unseen datasets. CLIP is pre-trained to predict if an image and a text snippet are paired together in its dataset. The authors use this capability to transform class labels into a text snippet describing the image for classification tasks. The predicted class is simply the one with the highest cosine similarity. Here we observe the usage of natural language to interface with the CLIP model in order to prepare it for different tasks in the zero-shot setting. This gives precedence to a new task for preparing CLIP for a zero-shot setting, **prompt engineering**.

Prompt engineering means designing a prompt that incorporates the label to be predicted in such a way that it gives good results when used with CLIP. As CLIP is trained on image-caption pairs scraped from the internet, it mostly encountered captions with multiple words describing the content of the image or the essence of the image. So for downstream tasks, the prompt must be created such that CLIP has the best chance of relating it with the image. So if an image contains a dog and the classification label is *dog*, the generated prompt would be: "A photo of a *dog*. The exact method they used for prompt engineering is "*The photo of a* $\langle label \rangle$." and it resulted in decent performances. This concept also follows the idea of engineering prompts for GPT-3 [59].

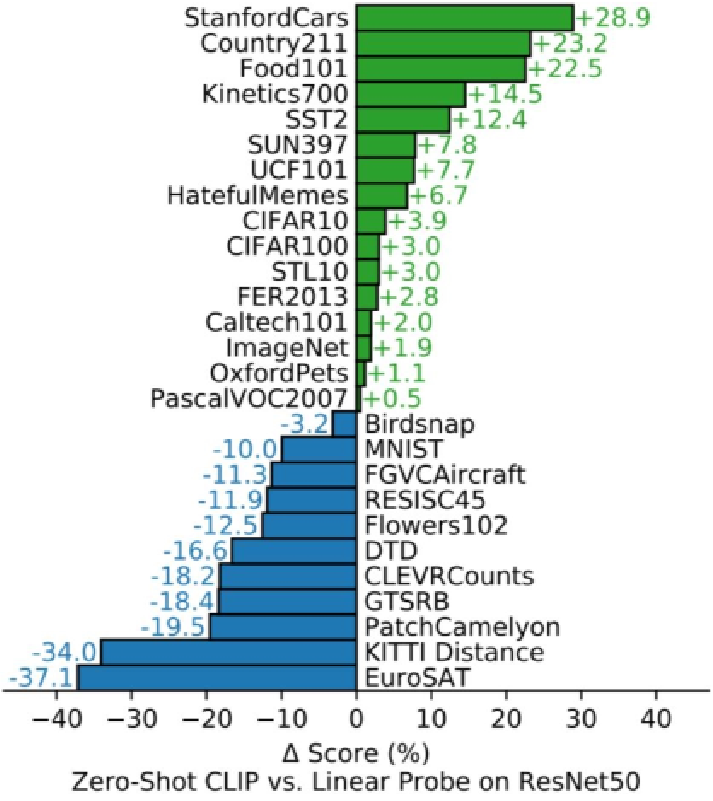


Figure 2.21: Zero-shot CLIP [9] demonstrates competitive performance compared to a fully supervised baseline. Evaluation across 27 datasets shows that the zero-shot CLIP classifier outperforms the fully supervised linear classifier.

Contributions:

1. Introduction of a novel approach for training visual models using alternative forms of data annotation.
2. Development of a large-scale dataset called Conceptual Captions, consisting of millions of image-caption pairs.
3. Utilization of natural language supervision instead of traditional annotations for images.
4. Introduction of Contrastive Predictive Coding (CPC), a self-supervised learning technique for learning effective visual representations.
5. Maximization of agreement between different image regions and their corresponding captions through CPC.
6. Improved generalization and transferability of learned representations to various visual tasks.

Limitations:

1. The reliance on the Conceptual Captions dataset may introduce biases or limitations inherent to the dataset itself, affecting the generalization of the trained visual models.
2. The use of natural language supervision introduces a level of subjectivity in the annotations, which may result in inconsistencies or inaccuracies in the training process.
3. The effectiveness of the proposed approach heavily depends on the availability and quality of textual descriptions associated with the images. In cases where such descriptions are sparse or of poor quality, the performance of the visual models may be compromised.

2.6.2 How Much Can CLIP Benefit Vision-and-Language Tasks? [10]

Shen *et. al.* [10] explored the idea of using CLIP [9] in vision-language tasks such as VQA for its excellent modality alignment. They observed that the majority of current Vision-and-Language Models (VLMs) for perceiving the visual environment rely on pre-trained visual encoders and use a relatively small set of manually annotated data (as compared to web-crawled data). However, large-scale pre-training usually results in

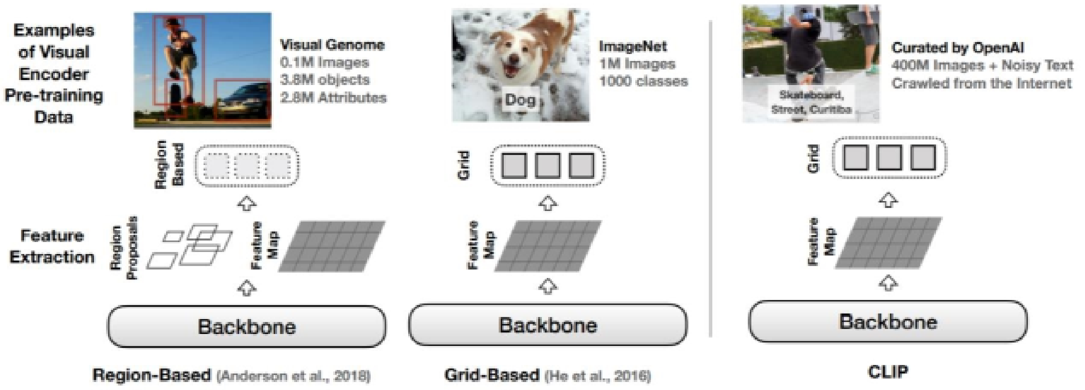


Figure 2.22: Comparing CLIP [9] to other visual encoders. Regional approaches are trained using [10] item detection data. Previous research uses either image classification or detection data for grid-based algorithms [2]. However, CLIP simply needs a text that is aligned. These encoders are swapped out by CLIP encoders.

better generalization performance, like CLIP, trained on a massive amount of image-caption pairs which has shown a strong zero-shot capability on various vision tasks. They explored if these capabilities could be transferred to vision-language understanding tasks such as Visual Question Answering, Visual Entailment, and V&L Navigation tasks. But transferring these capabilities is not that straightforward as CLIP does not contain a generator or decoder, which is often needed for vision-language tasks. One example could be image captioning. Without a generator, captions for an image cannot be generated from the representations extracted by CLIP.

The authors explore the usage of CLIP in two distinct scenarios. One way they changed the regular vision-language encoders with the CLIP encoders that learned to align vision-language examples. Here, no pre-training was needed and existing architectures already facilitated the transfer of CLIP to do vision-language tasks. The other scenario was combining CLIP with V&L pre-training and transferring to downstream tasks.

In the first case, the CLIP encoders were fine-tuned on Visual Question Answering and due to having a strong relationship between the visual and textual modalities, outperform traditional VQA methods like Pythia [74] and MCAN [49]. But it is not surprising as these methods of VQA are not the current state of the art. It is expected that CLIP with its superior embeddings of vision-language modalities would outperform these aging methods. The current methodology for VQA tasks is to use large vision-language models with extensive pre-training. The authors then tried to use CLIP in such a setting. Here, they took vision-language models, like LXMERT [51] and Pixel-BERT [75], which do not treat the different modalities in separate streams. These methods process vision-language inputs together and rely on multiple pre-training tasks on both modalities. CLIP-based encoders barely outperform these methods as both have strong image-language understanding.

In this work, the authors introduce a very simple zero-shot VQA method using CLIP as an interface between image and language modalities. Which is **QIP** (Question Invariant Prompting). Here the CLIP model is prompted the same way for every question. The prompt is rigid and it is like: *"Question: [The question] Answer:"*. Then CLIP is used to find the answer that best matches the image given the prompt. This method performs very poorly as CLIP was trained on matching image captions and captions are not usually like the prompt designed by the authors.

Contributions:

1. Extensive evaluation of the CLIP model across various vision-and-language tasks, such as image classification, object detection, and visual question answering.
2. Comparison of CLIP's performance against other state-of-the-art models specialized for individual tasks, highlighting its competitive performance.
3. Exploration of the generalization capabilities of CLIP by leveraging a large-scale dataset with image-text pairs, demonstrating its ability to achieve impressive results across diverse tasks.

Limitations:

1. The paper does not thoroughly explore the potential shortcomings or weaknesses of the CLIP model in specific vision-and-language tasks.
2. There could be factors not adequately addressed in the paper that may affect the generalization and performance of CLIP across different datasets or domains.
3. The evaluation of CLIP's performance may not consider certain complexities or challenges that could arise in real-world scenarios or niche applications.
4. The paper does not extensively discuss the interpretability or explainability of the CLIP model, which may be important in certain contexts.

2.6.3 CLIP Models are Few-shot Learners: Empirical Studies on VQA and Visual Entailment [11]

Song *et. al.* [11] picks up from Shen *et. al.* [10] and develops the idea of zero-shot VQA using CLIP. The main shortcoming of **QIP** was the rigid prompt. The authors decided to develop a method for generating variable prompts from the given question.

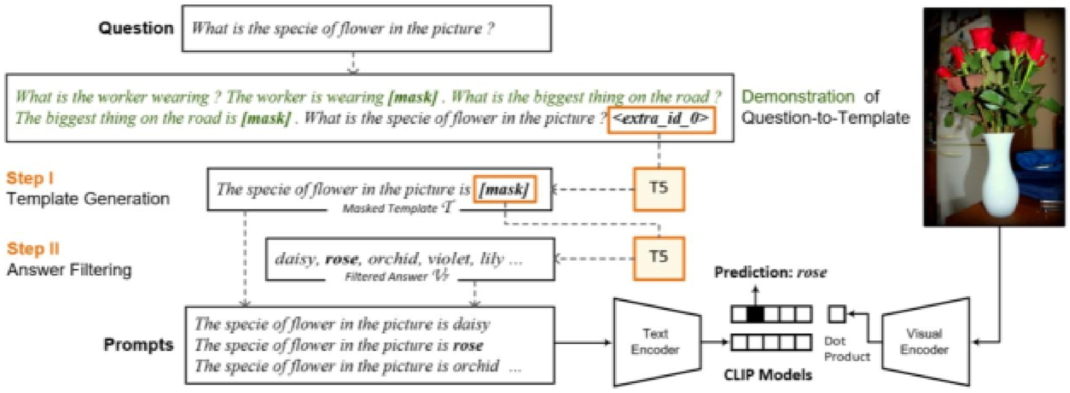

Question
What is the specie of flower in the picture ?
What is the worker wearing ? The worker is wearing [mask] . What is the biggest thing on the road ? The biggest thing on the road is [mask] . What is the specie of flower in the picture ? <extra_id_0>
Demonstration of Question-to-Template
Step I
Template Generation
The specie of flower in the picture is [mask]
Masked Template T
T5
Step II
Answer Filtering
daisy, rose, orchid, violet, lily ...
Filtered Answer V_F
T5
Prediction: rose
Prompts
The specie of flower in the picture is daisy
The specie of flower in the picture is rose
The specie of flower in the picture is orchid ...
Text Encoder
Dot Product
Visual Encoder
CLIP Models

Figure 2.23: The TAP-C [11] method proposes a framework for zero-shot Visual Question Answering (VQA). It involves generating a masked template from the question, filtering out impossible answers, generating prompts by infilling the template with selected answers, and using CLIP [9] to calculate image-text alignment scores for zero-shot VQA. The method combines language models and image-text alignment models to address the challenge of zero-shot VQA.

This would greatly benefit the model as CLIP was trained on image captions. This method depends on using a large language model like T5 [76] to turn the question into an answer template. The answer is *masked* as the model only transforms questions into answerable templates. Then another T5 model was used to answer the masked sentence. This generated multiple candidate answers. Then using CLIP, the candidate answers are matched with the images. The one with the maximum similarity is selected as the answer.

This method improved upon the zero-shot capabilities of CLIP for doing VQA. As the transformed questions are more in line with the image captions that CLIP is trained on, it outperforms **QIP**. The authors call this method **TAP-C** (Template-Answer Prompt then CLIP). Transforming the question into an answer template looks very similar to image captions. That is why, using a language model to fill in the answer templates creates pseudo-captions for those images. And CLIP excels in image-caption matching. Thus, **TAP-C** performs better than **QIP**. But one drawback of such a method is that, while generating the candidate answers using a large language model, the method treats it as a mask-filling task. The language model is totally unaware of the image and only outputs what is the most likely outcome given the answer template. Thus, if the image contains something that the language model deems unlikely, it will not generate an answer that contains that. By doing so, CLIP would also provide the wrong answer. If the language model was aware of the image and considered it when generating the candidate answer, then it would have been better. But large language models can process only a single modality. This is a limitation that is explored in later works.

Contributions:

1. The paper delves into the capabilities of the CLIP model in learning from a small number of labeled examples, shedding light on its potential for few-shot learning.

2. The authors showcase how CLIP performs admirably when compared to existing models on the tasks of Visual Question Answering (VQA) and Visual Entailment, even with limited training data.

3. The paper demonstrates CLIP's ability to transfer knowledge to new domains without extensive task-specific training, allowing it to answer questions and make entailment judgments effectively.

4. The authors investigate the influence of pretraining on CLIP's image and text modalities, emphasizing the significance of large-scale pretraining for robust few-shot learning performance.

5. Through careful analysis and experimentation, the paper provides valuable insights into the behavior of CLIP models, offering a deeper understanding of their strengths and limitations in few-shot learning scenarios.

Limitaions:

1. Factors such as question complexity, dataset biases, or domain-specific nuances may impact the accuracy and generalizability of the results.

2. While CLIP demonstrates impressive few-shot learning capabilities, it is still susceptible to biases present in the training data. Biases can lead to skewed performance, especially when encountering novel or underrepresented examples, highlighting the need for careful consideration of dataset biases during evaluation.

3. The paper does not explore the fine-grained control over CLIP's behavior for specific tasks. While CLIP's generalization abilities are impressive, there may be instances where task-specific fine-tuning or adjustments are necessary to achieve optimal performance in certain applications or domains.

4. The paper does not extensively address the scalability of CLIP models to large-scale datasets. As the dataset size increases, computational and memory requirements may pose challenges, potentially limiting the practical application of CLIP for real-world scenarios involving massive amounts of data.

5. The paper may not provide in-depth ablation studies exploring various design choices or model configurations. A more comprehensive exploration of different hyperparameters or architectural variations could provide valuable insights into the factors influencing the few-shot learning performance of CLIP models.

2.6.4 Plug-and-Play VQA: Zero-shot VQA by Conjoining Large Pre-trained Models with Zero Training [12]

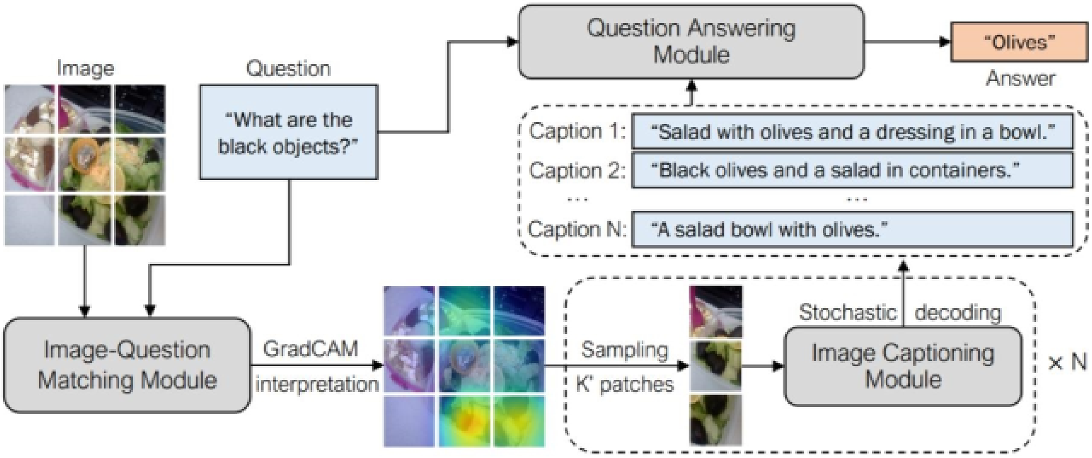


Image
Question
"What are the black objects?"
Question Answering Module
"Olives"
Answer
Caption 1: "Salad with olives and a dressing in a bowl."
Caption 2: "Black olives and a salad in containers."
...
Caption N: "A salad bowl with olives."
Stochastic decoding
Image-Question Matching Module
GradCAM interpretation
Sampling K' patches
Image Captioning Module
× N

Figure 2.24: The PNP-VQA system architecture [12] consists of three pre-trained modules: an image-question matching module, an image captioning module, and a question-answering module. The image-question matching module utilizes BLIP [3] to identify relevant image patches based on the question. The image captioning module generates diverse captions using BLIP. For question answering, the system adopts the UnifiedQAv2 model [18]. By combining these modules, PNP-VQA enables effective visual question answering by leveraging pre-trained models for image understanding and natural language processing.

Plug-and-Play VQA (PNP-VQA) is a modular framework designed for zero-shot Visual Question Answering (VQA). Unlike many existing approaches that necessitate extensive adaptation of pre-trained language models (PLMs) to incorporate visual information, PNP-VQA does not require additional training of PLMs. Instead, the authors propose the utilization of natural language and network interpretation as an intermediate representation that connects pre-trained models. The PNP framework begins by generating informative image captions guided by the corresponding questions. These captions are then used as context for question answering by passing them to a PLM. This approach allows us to effectively leverage the capabilities of pre-trained models without the need for additional training specific to the visual modality.

In order to ensure that the generated captions in PNP-VQA are specific and relevant to the question, the authors recognize that while an image contains abundant information, the question typically focuses on particular objects or regions within the image. Therefore, they force the model to create captions that go well with the image regions which are directly related to the question, rather than generic captions with no specific purpose. To achieve this objective, they use BLIP [3], a large-scale pre-trained vision-language model. BLIP includes a network branch known as the Image-grounded Text Encoder (ITE), which incorporates a vision transformer [65] for encoding the image and a textual encoder that attends to the image features through cross-attention. The ITE branch contains a similarity score, which measures the similarity between an image and a text description. To encode the image, the image encoder in ITE divides the image into k patches of equal size, enabling a comprehensive representation of the visual content. This division into patches allows the model to capture information from different parts of the image effectively.

Tiong *et. al.* [12] tackles the question of using large pre-trained models in multi-modal tasks, specifically VQA. Inspired by the recent astronomical development of large pre-trained language models (PLMs) such as [59, 68, 70, 71, 76], the authors inquired how to use these models in VQA. But these models lack multi-modal understanding ability. Thus, they decided to use network interpretability methods like GradCAM [19] as an interface between the visual and textual modalities. This method bridges the gap between the modalities and harnesses the potential of PLMs for multi-modal tasks such as VQA. The authors designed their method in a modular way so that anyone can replace the PLM module or the image processing module for better alternatives.

Their framework for zero-shot visual question answering which conjoins large pre-trained models with zero additional training is called **PNP-VQA: Plug and Play VQA**. They first take the image and divide it into multiple patches. Then they take the question and perform the GradCAM operation on the image conditioned by the question. This gives k patches where it matters the most for answering the question. After the k patches are selected, they are sent to an image captioning module. This generates n captions from the k patches. Here, the authors used BLIP [3] to generate captions for the selected image patches. They call this *"Question Guided Captioning"*. Their work is a derivative of PiCA [77] where they generate a single caption from the image and use it as a context for answering the question. After generating the captions, the authors combine them after encoding them. It is called *Fusion-in-Decoder* or **FiD**. Then the embeddings are sent to a PLM which answers the question.

An image serves as a rich source of information, but the question at hand is likely focused only on particular objects or regions [12]. Therefore, the authors encourage PNP-VQA to generate captions that describe image regions relevant to the question instead of generic captions with no specific aim [12]. They accomplish this goal by leveraging BLIP [3], a large-scale pre-trained vision-language model that contains a

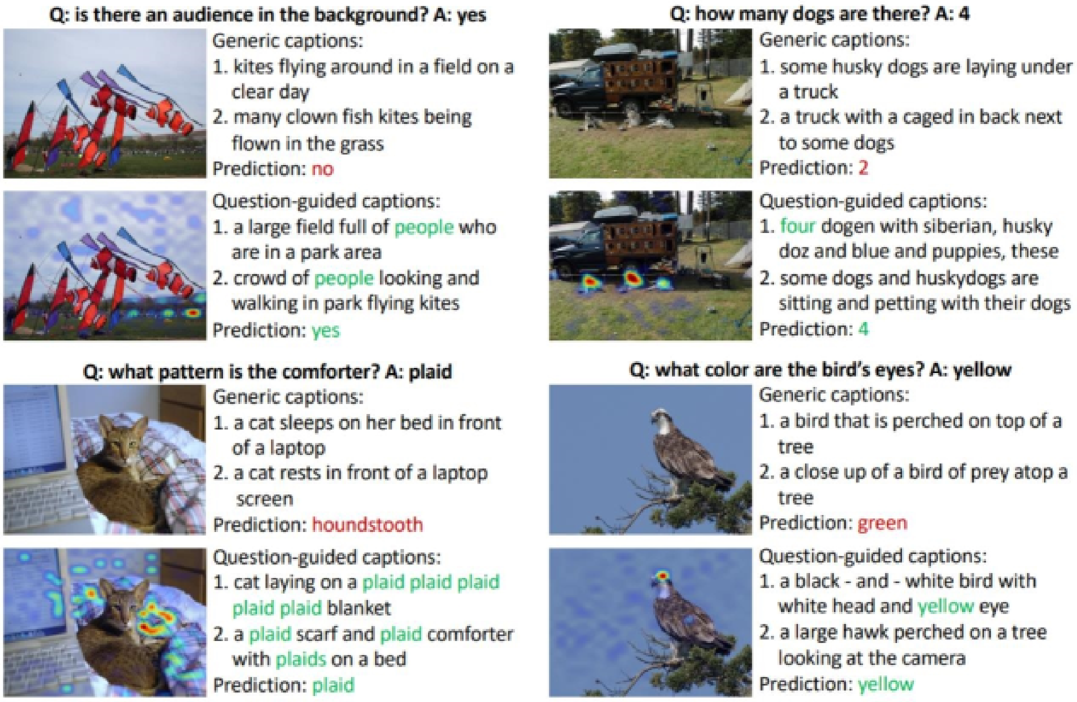

Q: is there an audience in the background? A: yes
Generic captions:
1. kites flying around in a field on a clear day
2. many clown fish kites being flown in the grass
Prediction: no
Question-guided captions:
1. a large field full of people who are in a park area
2. crowd of people looking and walking in park flying kites
Prediction: yes
Q: how many dogs are there? A: 4
Generic captions:
1. some husky dogs are laying under a truck
2. a truck with a caged in back next to some dogs
Prediction: 2
Question-guided captions:
1. four dogen with siberian, husky doz and blue and puppies, these
2. some dogs and huskydogs are sitting and petting with their dogs
Prediction: 4
Q: what pattern is the comforter? A: plaid
Generic captions:
1. a cat sleeps on her bed in front of a laptop
2. a cat rests in front of a laptop screen
Prediction: houndstooth
Question-guided captions:
1. cat laying on a plaid plaid plaid plaid plaid blanket
2. a plaid scarf and plaid comforter with plaids on a bed
Prediction: plaid
Q: what color are the bird's eyes? A: yellow
Generic captions:
1. a bird that is perched on top of a tree
2. a close up of a bird of prey atop a tree
Prediction: green
Question-guided captions:
1. a black - and - white bird with white head and yellow eye
2. a large hawk perched on a tree looking at the camera
Prediction: yellow

Figure 2.25: We can generate two types of captions for VQAv2 [15] data: generic captions that describe the entire image, and question-guided captions that focus on specific image patches. The question-guided captions are generated by using GradCAM [19] heatmaps to identify relevant patches. These different types of captions provide varied perspectives and insights into the image, allowing for a more comprehensive understanding of the visual content in VQAv2 scenarios.

network branch outputting a similarity score $sim(v, t)$ between an image v and a text t. This branch, called Image grounded Text Encoder (**ITE**), employs a vision transformer [65] that encodes the image, and a textual encoder that attends to the image features using cross-attention. The image is evenly divided into k patches before being fed into the image encoder. We send the picture v and the question t to the ITE network in order to find important image patches. We next use a variant of GradCAM, a feature-attribution interpretability technique, which aggregates all cross-attention mappings using gradient weights. The cross-attention scores, $A \in R^{M \times K}$, can be written as:

$$A = softmax\left(\frac{YW_QW_K^TX^T}{\sqrt{D_t}}\right) \tag{2.2}$$

Then GradCAM computes the partial derivative of the similarity score from ITE with respect to the cross-attention scores. Then multiplies the gradient matrix element-wise with the cross-attention scores and computes a weighted average across the heads. This would be like:

$$rel(i) = \frac{1}{H}\sum_{j=1}^{M}\sum_{h=1}^{H} min \quad 0, \frac{\partial sim(v,t)}{\partial A_{ji}^{(h)}}\Bigg) A_{ji}^{(h)} \tag{2.3}$$

Here, h denotes the number of attention heads. The attention matrix A can be used to determine the significance of a patch. Inspired by GradCAM [19], they eliminate attention scores that aren't useful by multiplying them by the gradient, which could make the image-text similarity grow.

The authors deduced that using GradCAM gives them the ability to pick out which image patches are necessary for answering a question according to the question. This idea is remarkable for being modular and finding a simple yet effective way to conjoin PLMs with existing image processing methods. Using network interpretability methods to choose relevant image patches for captioning could be improved upon by using special attention mechanisms. We discuss it in the following literature review.

Contributions:

1. The paper introduces an innovative method for Visual Question Answering (VQA) that does not require task-specific training. It proposes a framework that combines existing pre-trained models from the language and vision domains, enabling VQA without the need for extensive training.

2. The authors utilize the knowledge captured by pre-trained models, including language models and visual models, to generate answers to questions about images. This approach taps into the general understanding of these models, allowing them to answer questions without specialized training.

3. The paper suggests a two-component architecture that integrates language and visual models. This combination enables the system to generate plausible answers using the language model and rank them based on visual compatibility using the visual model. This integration facilitates zero-shot VQA without requiring specific training.

4. The paper presents experimental results on established VQA datasets, comparing the proposed framework to fully trained models. The results demonstrate competitive performance despite the absence of task-specific training, indicating the effectiveness of the approach for zero-shot VQA.

5. The contributions of the paper have implications for various practical applications where VQA is needed but limited or no task-specific training data is available. The proposed framework opens up possibilities for deploying VQA systems in real-world scenarios without extensive training requirements.

Limitations:

1. The effectiveness of the approach heavily relies on the quality and comprehensiveness of the pre-trained language and vision models used. Limitations in these models, such as biases or knowledge gaps, can impact the accuracy of the generated answers.

2. The plug-and-play framework might struggle to adapt to highly specialized or domain-specific VQA tasks. The lack of task-specific training can result in suboptimal performance when the questions and images deviate significantly from the data the pre-trained models were exposed to.

3. The pre-trained models utilized in the framework may inherit biases present in their training data, which can manifest in the generated answers. Additionally, there is a risk of overgeneralization or undergeneralization, where the system might provide overly generic or overly specific answers due to the lack of task-specific training.

4. As the size of VQA datasets grows, the plug-and-play framework may encounter scalability challenges. The computational resources required to process and integrate information from large-scale datasets can become prohibitive without task-specific training or optimizations.

Chapter 3

Visual Robustness Analysis for VQA

In this chapter, we discuss the background knowledge for our framework followed by the related definitions that pave the way for the next section – the proposed visual robustness evaluation framework. We further define a new set of evaluation metrics and delve into the experimental results of our framework.

3.1 Background Study

In the previous chapter, we described traditional robustness approaches along with robustness works on image classification [7] and on texts of VQA systems [6]. CARETS [8] proposed a test suite along with a dataset and new metrics for both robustness and consistency. [6] also proposed new robustness metrics along with a basic question set for robustness evaluation. However, the work primarily focused on textual robustness without considering corruptions or noise on the image input.

At this point, it can be established that there is an absence of literature on the visual robustness of VQA models. Furthermore, there is no metric to quantify a model's robustness or quantify the effect of a particular form of corruption on the model's performance. To deploy a VQA model in a real-life test environment, one must be able to quantify how prone the model is to image corruption or visual effects that commonly occur in real-life as image corruption is more common than textual corruption. Oftentimes, images taken from live-cam and video feed will be used for VQA which might degrade the performance of models with high accuracy.

[7] explored real-life corruption effects on classification networks but their work has some major flaws. Firstly, the work is outdated, and hence, their results are useless as modern architectures evolved drastically. While most of their works focused on neural networks, the world is shifting towards transformers-based [78] architectures. Secondly, they do not propose a complete framework, and hence, the outdated models cannot be replaced by newer ones. Finally, and most importantly, their methodol-

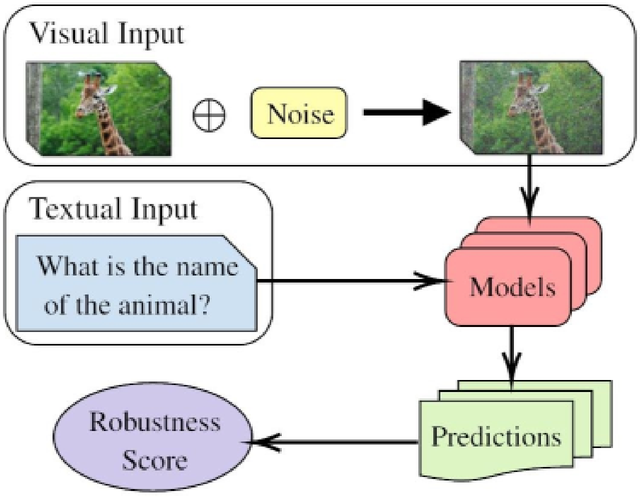


Figure 3.1: Overview of the Visual Robustness Framework: noise is applied to the visual input and then fed to a set of models with unaltered textual input. The robustness score is calculated from the resulting set of predictions.

ogy cannot be generalized to VQA. While their corruption effects can be viewed as the template for visual corruption on VQA images, we can also assert that there is a substantial difference between working with images for image-only classification networks and working with images on image-text VQA networks.

3.2 Related Definitions

Based on the previous discussion, we will dive into the quantification of visual robustness by first defining visual robustness for VQA models and then defining the corruption functions that are common occurrences in practical scenarios.

3.2.1 Defining Visual Corruption

We define visual corruption as the degradation of the image caused by various factors such as noise, blur, or compression artifacts. Our work primarily focuses on the prevalent forms of degradation that are commonly observed in real-world scenarios. Our definition of visual corruption is similar to [7] which is substantially distinguishable from adversarial attacks. The associated metric to test the robustness does not consider adversarial attacks either.

We first define our VQA model as a classifier $f : \mathcal{X}_I, \mathcal{X}_Q \rightarrow \mathcal{Y}$ and the training samples are from the joint probability distribution are $\mathcal{D}_{I,Q}$. We further define a set of functions that perform visual corruption as $\mathcal{C}$ where each function is associated with a probability of occurring in real life as $\mathcal{P}_{\mathcal{C}}(\tau) \in \mathbb{P}_{\mathcal{C}}$. We can now define the visual robustness $\mathbb{B}$ of our classifier on the test dataset which follows the same distribution $\mathcal{D}_{I,Q}$ such that,

$$\mathbb{B}_{\tau\sim\mathcal{C}}[\mathbb{P}_{(x_i,x_q,y)\sim D_{I,Q}}(f(\tau(x_i,x_q)=y))] \tag{3.1}$$

which corresponds to the average-case performance making it a suitable metric for robustness. On the contrary, adversarial attacks correspond to the worst-case performance on the classifier's response to small and specific changes made to the input data, and such changes are tailored to that specific classifier instead of a corruption function that affects classifiers universally. The adversarial robustness can be defined as

$$min_{||\delta||_p<\epsilon}[\mathbb{P}_{(x_i,x_q,y)\sim D_{I,Q}}(f(x_i+\delta_i,x_q+\delta_q)=y)] \tag{3.2}$$

where ϵ is a small increment. Since we are more concerned about the performance of our models on real-world corruption effects, we would hence define robustness as the average-case performance instead of the traditionally defined adversarial robustness which deals with the worst-case performance.

3.2.2 Visual Corruption Functions

At the time of writing this paper, our framework comprises 17 visual corruption functions, analogous to image processing functions inspired by [7, 79], which are used to create a diverse set of artificially corrupted image datasets. *Most* of the corruption functions have severity levels starting from severity level - 0 i.e. the uncorrupted image and each level is defined by a set of parameter values set in a way that the difference between two consecutive severity levels is noticeable to a human observer. The transformation functions are chosen to replicate corruption effects that resemble realistic conditions. We have categorized the set of corruption functions into 4 broad categories similar to [79].

Arithmetic Noise

Arithmetic noise modifies the image by performing arithmetic operations e.g. addition, multiplication, negation, etc on all of the color channels. A subcategory of arithmetic noise is **Additive Noise** the adds a particular value that comes from a distribution $\mathcal{D}$ to every pixel in the image. Additive noise is implemented in the form of **Gaussian Noise** and **Poisson Noise**. Gaussian Noise appears in low-light conditions [7] and is one of the most frequently occurring types of noise in telecommunications and digital image [80]. Poisson Noise, often referred to as Shot Noise, occurs due to the nature of light behaving as a quantized particle [81]. To define additive noise, We first define the random variable X as $X_n \sim N(\mu, \sigma^2)$ where the probability distribution N is defined as $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and $X_p \sim P(\lambda)$ where the probability distribution P is defined as $f(x) = \frac{\lambda^x}{x!} e^{-\lambda}$. Hence, the transformation function for additive noise can be defined as $\mathcal{T}(r) = r + Y$ where $Y = X_n$ for Gaussian Noise and $Y = X_p$ for Poisson noise. The severity levels are defined by changing the parameter values of the aforementioned probability distributions.

Another subcategory of arithmetic noise is **Multiplicative Noise** implemented in the form of **Speckle Noise** which is defined as $\mathcal{T}(r) = r + r * X_n$ where X_n is the random variable $X_n \sim N(\mu, \sigma^2)$ as defined earlier. Speckle noise is a common occurrence in medical and radar images [82]. **Color Inversion**, a common digital image processing operation, simply negates the image, $\mathcal{T}(r) = -r$, and has a single severity level.

Value Assignment Noise

As the name suggests, value assignment noise has a probability p of assigning a particular value to a pixel, $\mathcal{T}(r) = k$, on all of the color channels. This noise is primarily implemented in the form of **Impulse Noise** which is typically one of the two types - bipolar impulse noise, commonly known as **Salt and Pepper Noise**, and **Random Valued Impulse Noise**. Salt and pepper noise takes one of two values, typically between the maximum intensity value r_{max} and the minimum intensity value r_{min} – each with an equal probability p of occurrence. Random-valued impulse noise takes a particular value from a range of values, typically $[r_{min}, r_{max}]$, and follows a uniform distribution for the probabilistic occurrence of the values. A defective camera sensor might cause impulse noise during capturing and transmitting the image [80, 83]. Another form of value assignment can take place in the form of **Thresholding** i.e. the pixel will be assigned a particular value if it exceeds or subceeds a particular threshold value. **Binary Thresholding** is defined as $\mathcal{T}(r) = r_{max}$ if $r > r_{thresh}$, otherwise, $\mathcal{T}(r) = r_{min}$.

Image Attribute Transformation

An image has several attributes e.g., brightness, saturation, and color property, which can be modified by the image attribute transformation functions. Our framework includes five transformation functions – Brightness, Contrast, Saturation, grayscale, and Grayscale Inversion. To modify the **Brightness**, we transform the image from the RGB color model to the HSV color model and add a constant value to the *value* channel of the HSV image resulting in the increase of brightness and intensity of the image. By adding a negative value to the *value* channel, the function will darken the image. The function can be defined as $\mathcal{T}(v) = v + c$ where v represents the value of the **value** channel and c represents the additive constant. In real-life scenarios, lighting effects, luminance adjustment in digital displays, photographic effects, and other factors can cause an image to appear brighter or darker. By simulating these effects using the brightness function, our framework can be used to test the robustness of VQA models to varying lighting and display conditions.

Saturation refers to the purity of the colors in an image [84] and can be used to enhance the quality of the image i.e. the image will look visually appealing to a human observer. However, oversaturation might make the image look artificial to an observer, and undersaturation might produce washed-out effects that can adversely affect the image quality. Changing the saturation of an image is common in digital image processing to make the image look aesthetically pleasing. To change the saturation, the image is transformed from RGB to HSV color model, and the *saturation* channel value is modified by multiplying and adding a constant value i.e.$\mathcal{T}(v) = v * c_1 + c_2$ where c_1 and c_2 represents the multiplicative and additive constants respectively which are set based on the severity of the noise.

Contrast refers to the difference of color intensity values between different parts of the image [84] i.e. how well the details of an image are distinguishable. Usually, high contrast is more appealing to a viewer as it sets clear boundaries between various color intensities. On the contrary, low contrast causes difficulty in differentiating the details and hence, producing washed-out effects. Contrast enhancement is a common image-processing technique applied to spatial, frequency, and wavelet domains using contrast stretching, histogram equalization, etc. The contrast changing is defined as, $\mathcal{T}(r) = (r - \mu_{H,W}) * c + \mu_{H,W}$ where $\mu_{H,W}$ represents the average pixel intensity over the *height* and *width* channel and c represents the multiplicative constant. For all 3 transformation functions, the output values are clipped from 0 to 1.

Grayscale can be thought of as a transformation function that modifies the color property of the image. Grayscale works by simply averaging the pixel values over the color channels i.e. $\mathcal{T}(r) = \mu_C$ where μ_C represents the average pixel intensity over the *color* channel. **Color Inversion** has been discussed as arithmetic noise but it can also be classified as an attribute transformation function since it modifies the color property of an image. **Grayscale Inversion** is simply the combination of grayscale and color inversion; defined as $\mathcal{T}(r) = -\mu_C$. grayscale images are common in real-life systems where it is not possible to represent the color information of a digital image. Several systems like medical imaging, document scanning, security systems, etc. use grayscale images, and systems like night vision, medical imaging, astronomy, etc. use color-inverted images. Thus, it is necessary for VQA models to perform well on both grayscale and color-inverted images.

Blur Noise

Blur noise deals with blurring effects similar to convolving with an averaging filter and can be mathematically described as $\mathcal{T}(\mathcal{I}) = \mathcal{I} * K$ where $\mathcal{I}$ represents the digital image and K represents the kernel and convolution operation for a 2D image is defined as

$$(f * g)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) \cdot g(x - i, y - j)$$

While Gaussian blur and median blur are common blurring functions, we shall define 3 other realistic blurring functions that have common real-life applications. **Defocus Blur** performs channel-wise convolution, and the function is defined as $\mathcal{T}(\mathcal{I}) = \mathcal{I} * K_{r,a}$ where $K_{r,a}$ is a disk-kernel with r radius and a alias blur. The constants r and a vary across the severity levels. Defocus blur replicates the blurring effect in cameras where the subject is out of focus. **Zoom Blur** occurs due to rapid camera motion towards an object and **Frosted Glass Blur** imitates the appearance of an object while looking through frosted glass [7]. Both effects do not have strict definitions and our framework follows the implementation from [7].

Miscellaneous Effects

Apart from the previous transformation functions, we add weather effects which try to make the image appear in a particular weather condition. At the time of writing this paper, our framework includes only the **Snow Effect** but we wish to include other effects like fog, frost, rain, and clouds in the future. Some transformation functions try to create physical effects on the images. The **Splatter Effect** makes the image look like it has been splattered by paint or any form of liquid. The **Elastic Effect** simulates the effect of stretching or wrapping the image. Finally, we include a couple of transformation functions that replicate digitization effects. The **Pixelate Effect** is a visual effect that creates a mosaic-like appearance similar to visible image pixels appearing due to lower resolutions. The function works by replacing smaller details and textures with larger blockier shapes. Pixelation is commonly used for stylistic purposes and for censoring parts of an image. **JPEG Compression Effect** tries to emulate the loss of image information due to JPEG compression. Similar to the previous category, all the miscellaneous functions do not have strict definitions, and our framework's implementation follows [7].

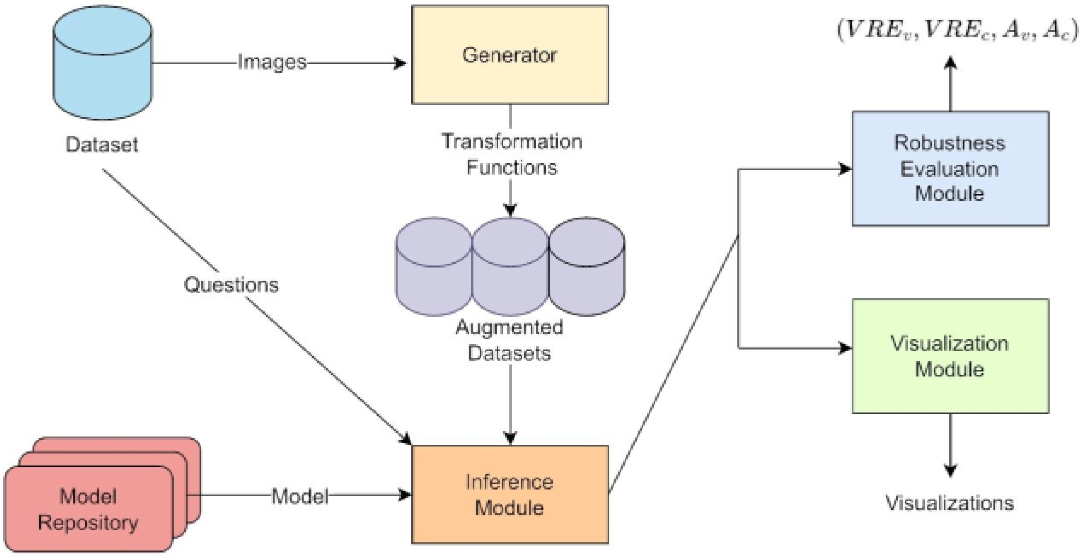


Figure 3.2: Architecture of the proposed Visual Robustness Framework.

3.3 Proposed Methodology

We created a framework for evaluating the visual robustness of current and future VQA models. Our framework is designed to be modular and resilient. It allows users to utilize their own datasets and models for evaluation of performance, as well as implement their own corruption functions for their special needs. By doing so, our framework provides greater flexibility to researchers and practitioners to test the visual robustness of any model using various data and scenarios. Fig-3.2 shows the primary elements of our framework that take a dataset and a collection of models as inputs and produce the VREs, accuracy scores, and visualizations as output. The robustness and accuracy metrics in our framework have been discussed in the previous section. The framework consists of a model repository, a generator module, an inference module, a robustness evaluation module, and a visualization module.

The images of the dataset are passed to the generator module which produces the augmented dataset using a set of transformation functions defined earlier. Our framework is completely modular and hence, the users have the flexibility to add or remove transformation functions if necessary. Most of the transformation functions will have 6 severity levels; the first one being the uncorrupted dataset. The impact of *strength* of the noise is proportional to the value of the severity level i.e. it will visually get tougher to understand the context from the image with higher severity levels.

As the first severity level, i.e. level - 0, is the same for all datasets, for n such transformation functions, we will have $5n + 1$ augmented datasets (the uncorrupted dataset is also considered as part of the augmented dataset). The model repository will pass a model which will run inferences on the augmented datasets. The output will be given in the format of JSON files which will be passed to the robustness evaluation and visualization module. All the robustness and accuracy metrics will be calculated by the robustness evaluation module and the results will be also be stored as a JSON file. The visualization module will simply create and store different forms of visualizations that we will be shown in the latter sections.

We designed our approach to track the performance of the models and report any anomalies detected during the experiment. The logger file provides information on how the model performed and stores errors such as any issues with the JSON files, running the models, etc. The JSON files produced by the inference module contain the questions that the model predicted correctly and actual answers taken from the dataset's annotation file.

Moreover, our framework currently comprises 17 visual corruption functions which are modular and expandable. Functions were chosen based on the category of corruption, the impact of noise on visual inference for answer generation, and the resemblance of the effect to realistic noise. The visual corruptions effects are similar to [7] with 5 levels excluding the base level but functions such as grayscaling and color inversion has a single level only. The transformations provide greater diversity in the dataset, making it more challenging for the models to accurately answer the questions.

3.4 Proposed Evaluation Metrics

We introduced several evaluation metrics in this work; the two primary metrics being the accuracy metric and our proposed Visual Robustness Error (VRE) metric. The Visual Robustness Error is an aggregated metric that tries to quantify error for a model or a particular type of corruption. Hence, higher VRE values for a model indicate that the model is *relatively* more prone to noise. We emphasize the word *relatively* as VRE performs a comparative normalization on the metric values and thus, the generated VRE scores are relative to the set of models used; the scores are not universal. Such a design choice has been made to use the framework as a comparative module which helps in comparative analyses and visualizations. However, users have the flexibility to replace the relative VRE with absolute VRE scores but our experimental results in this paper will not use the absolute VRE scores. In this work, when we refer to VRE, it indicates the relative VRE score specifically.

3.4.1 Model Evaluation

The robustness of a particular model depends on the type and severity of the visual corruption. As it is difficult to use a single pre-existing metric to estimate the robustness of a particular model or quantify the effect of a corruption type, we propose a better measure of robustness as an aggregation of metrics. While aggregating the metrics, we ensure the scores are *corruption-independent* and *model-independent* for model-based and corruption-based aggregations respectively. Furthermore, the average accuracy of the models and visual corruptions are also calculated and the corresponding error is integrated into the unified robustness score.

3.4.2 Accuracy Evaluation Metrics

The standard metric [21] to evaluate a VQA model's accuracy for open-ended tasks in VQAv2 dataset [15] for a particular model v, corruption type c, severity level l is defined as:

$$A^v_{c,l} = \frac{1}{N_Q}\sum_{q\in\mathbb{Q}} min \quad \frac{\sum_{a\in\mathbb{A}_q}\mathbb{I}[\hat{a}_q = a]}{3}, 1\Bigg) \tag{3.3}$$

where N_Q is the number of questions in the dataset, q is a particular question from the question set $\mathbb{Q}$, a is a particular human annoted answer to the question q from the corresponding answer set $\mathbb{A}_q$, $\mathbb{I}[\cdot]$ is the indicator function, and $\hat{a}$ is the answer predicted to by the model for q. We can now compute the average accuracy as:

$$A^v_c = \frac{1}{L}\sum_{l\in\mathbb{L}} A^v_{c,l} \tag{3.4}$$

where L is the number of severity levels, and l is a particular severity level from severity level set $\mathbb{L} = \{0, 1, 2, ...\}$. We further define average accuracy for a particular model as:

$$A^v = \frac{1}{C}\sum_{c\in\mathbb{C}} A^v_c \tag{3.5}$$

where C is the number of visual corruptions, and c is a particular visual corruption from the set of corruptions $\mathbb{C}$. Similarly, we define accuracy for a particular corruption function:

$$A^c = \frac{1}{V}\sum_{v\in\mathbb{V}} A^v_c \tag{3.6}$$

where V is the number of models, and v is a particular model from the set of models $\mathbb{V}$. It is to be noted that, the higher the accuracy for the corruption type, the *weaker* the corruption type is i.e. the corruption accuracy is inversely proportional to the *strength* of the corruption.

3.4.3 Robustness Error

As per the definition of accuracy 3.3, we can simply define the robustness error or simply, error as:

$$E^v_{c,l} = 1 - A^v_{c,l} \tag{3.7}$$

It should be noted that there can be multiple ways to define the error but we chose to use misclassification error due to its simplicity, usability, and similarity to the accuracy metric. According to [21], an answer is stated to be fully accurate if at least 3 human annotators gave the same answer for that particular open-ended question. By the definition of error, that answer will be *error-free*. Similarly, for 1 and 2 human annotated answers, there will be partial error, and for no annotated answers, there will be the maximum error. Similar to average accuracy, the error is also expressed in percentages.

The robustness error typically increases with the increase in corruption severity but experimental results *sometimes* showed a decrease in error with the increase in severity. This is valid by definition as a higher severity level is associated with the parametric values of the corruption function that visually produce a stronger corruption effect and such an effect might not always result in the deterioration of a model's performance. For instance, the image produced by the corruption function for the severity level - 5 will visually appear to have a stronger corruption effect when viewed by a human than the corresponding severity level - 4. However, the effect might result in some minor visual details being overlooked by the model which might help in answering certain types of questions analogous to having a *bird's eye view* on the image. Another example – pixelating an image to a greater degree will result in the model overlooking/ignoring the structural details of the image and might help answer color-related questions. Similarly, the corruption effect might also result in the model noticing/focusing certain details about the image e.g., brightening the image will help the model notice the darker details and plausibly result in more accurate answers to the related questions.

The relatively rudimentary VQA models exhibit this property of utilizing the corruption effect to increase the accuracy as they have a very high base error. Unbalanced and small-scale datasets might also result in undesired outcomes as such datasets aren't suitable for evaluation. We disregard this issue in our experimental results by using modern VQA models [4, 85] and a standard dataset [15]. However, for theoretical purposes, the maximum and minimum error is explicitly differentiated from the error values associated with severity levels 5 and 0 respectively.

3.4.4 Severity Aggregation Metrics

The severity levels are associated with the parametric values of a particular corruption type and our aggregation metric denoted by $\mathcal{M}_c^v$ where $\mathcal{M}$ is a generalized aggregation metric will combine the error values for a particular classifier v and for a corruption type c over the severity levels. As the severity levels are not strictly defined by the parametric values, these values are disregarded during aggregation.

First-Drop

The score is defined as the relative change of error when the least amount of corruption is applied i.e. the relative difference between the first severity level error and the uncorrupted/level-0 error. The rationale behind using the relative difference instead of the difference only is that the difference is dependent on a model's severity-0 accuracy or base accuracy while the relative difference doesn't have such dependencies. The word *drop* is used to signify the drop/decrease in accuracy due to the initial corruption level but when calculating robustness error, an increase in error is expected. Higher first-drop scores indicate that the model is more prone to the minor impact of a particular corruption. Mathematically, first-drop is represented as:

$$\mathcal{F}_c^v = \frac{E_{c,1}^v - E_{c,0}^v}{E_{c,0}^v} \tag{3.8}$$

Range of Error

The range of error indicates how spread out the error values are across severity levels and is calculated by taking the difference between the maximum and minimum error values; defined as:

$$\mathcal{R}_c^v = \underset{l}{max}[E_{c,l}^v] - \underset{l}{min}[E_{c,l}^v] \tag{3.9}$$

A higher range of error indicates a model's continuous decline of performance with increasing severity levels and the extent of the effect of a particular corruption type.

Error Rate

The change of error between severity levels is defined by the error rate and is calculated by taking the slope of the best-fit line satisfying the error values over severity levels.

We simply use the regression equation to define this metric:

$$\rho_c^v = \frac{L \sum_{l \in \mathbb{L}} \left(l E_{c,l}^v \right) - \left(\sum_{l \in \mathbb{L}} l \right) \left(\sum_{l \in \mathbb{L}} E_{c,l}^v \right)}{L \sum_{l \in \mathbb{L}} l^2 - \left(\sum_{l \in \mathbb{L}} l \right)^2} \tag{3.10}$$

Average Error

The average error is analogous to (3.4):

$$\mu_c^v = \frac{1}{L} \sum_{l \in \mathbb{L}} E_{c,l}^v \tag{3.11}$$

[7] uses a form of averaging known as corruption error that can be inferred as the average error of the current model relative to the average error of a base model for that particular type of corruption.

Average Difference of Corruption Error

A more nuanced metric inspired by [7] that aggregates error differences with the uncorrupted image and is defined as:

$$\Delta_c^v = \frac{1}{L-1} \sum_{l \in \mathbb{L}'} [E_{c,l}^v - E_{c,0}^v] \tag{3.12}$$

where $\mathbb{L}'$ is defined as the set of corrupted severity levels $\mathbb{L}' = \{1, 2, 3, ...\}$

3.4.5 Visual Robustness Error (VRE)

VRE is a unified score for a particular classifier v or for a particular corruption type c that aggregates the previously defined metrics $\mathcal{M}_c^v$ by calculating a weighted average of the normalized metric values.

Corruption Aggregation Metrics

Aggregating metrics for different types of corruption is more challenging than aggregating severity levels, as the difficulty level varies across the corruption types. We aggregate the generalized metrics $\mathcal{M}_c^v \in \mathbb{M}$ where $\mathbb{M} = \{\mathcal{F}, \mathcal{R}, \rho, \mu, \delta\}$ as

$$\mathcal{M}^v = \frac{\sum_{c \in \mathbb{C}} \mathcal{M}_c^v}{\frac{1}{V} \left(\sum_{v \in \mathbb{V}} \sum_{c \in \mathbb{C}} \mathcal{M}_c^v \right)} \tag{3.13}$$

The metric aggregates the corruption types $c \in \mathbb{C}$ where $\mathbb{C}$ denotes our set of corruption functions for a model $v \in \mathbb{V}$ where $\mathbb{V}$ denotes our set of VQA models and V denotes the number of models. The corruption aggregated metric gives us a value relative to the average metric value of the models and based on the value we can infer a model's robustness based on a particular metric. For example - if a particular model has a corruption-aggregated first-drop score greater than 1, it indicates that the model has an above-average first-drop score. Similarly, a score lesser than 1 indicates a more robust model.

Model VRE

VRE for a particular model v is defined as the weighted average of the corruption aggregation metrics:

$$VRE_v = \sum_{\mathcal{M} \in \mathbb{M}} W_{\mathcal{M}} \mathcal{M}^v \text{ where } \sum_{\mathcal{M} \in \mathbb{M}} W_{\mathcal{M}} = 1 \tag{3.14}$$

We introduce a hyperparameter α which is the weight assigned to the average error. As the summation of the weights will be equal to 1, the rest of the weights will be equally distributed between the other metrics. The rationale behind assigning a weight to the average error will be discussed in a latter section. The equation can be written as:

$$VRE_v = \left(\frac{1-\alpha}{4}\right)\mathcal{F}^v + \left(\frac{1-\alpha}{4}\right)\mathcal{R}^v + \left(\frac{1-\alpha}{4}\right)\rho^v + \alpha\mu^v + \left(\frac{1-\alpha}{4}\right)\delta^v \tag{3.15}$$

Model Aggregation Metrics

Similar to the corruption aggregation metrics, we aggregate the generalized metrics $\mathcal{M}_c^v \in \mathbb{M}$ for different models and define the model aggregation metrics as:

$$\mathcal{M}_c = \frac{\sum\limits_{v \in \mathbb{V}} \mathcal{M}_c^v}{\frac{1}{V}\left(\sum\limits_{c \in \mathbb{C}} \sum\limits_{v \in \mathbb{V}} \mathcal{M}_c^v\right)} \tag{3.16}$$

Corruption VRE

We also define the Visual Robustness Error (VRE) for a particular corruption function $c \in \mathbb{C}$ as:

$$VRE_c = \sum_{\mathcal{M} \in \mathbb{M}} W_{\mathcal{M}} \mathcal{M}^c \text{ where } \sum_{\mathcal{M} \in \mathbb{M}} W_{\mathcal{M}} = 1 \tag{3.17}$$

Similarly, the equations are formulated using α:

$$VRE_c = \left(\frac{1-\alpha}{4}\right)\mathcal{F}^c + \left(\frac{1-\alpha}{4}\right)\mathcal{R}^c + \left(\frac{1-\alpha}{4}\right)\rho^c + \alpha\mu^c + \left(\frac{1-\alpha}{4}\right)\delta^c \quad (3.18)$$

3.5 Performance Evaluation

In this section, we shall primarily look at the prerequisites to conduct experiments using our framework. Before analyzing the results, we will go through the setup, the dataset, and the models used.

3.5.1 Experimental Setup

We test the performance of six recent VQA models on the random subset of the validation split of the VQAv2 dataset [15], a large-scale dataset for VQA tasks that is considered a standard. The augmented datasets are produced using 17 transformation functions, but the result analysis on grayscale and grayscale inverse have been omitted. When an image is converted to a grayscale image or inverted, it loses its color information. Afterward, the model will not be able to answer color-related questions. Again, if the model accurately answers a color-related question i.e., its answer matches the annotated answer, then that means the model associates the color with a particular shape, and hence, exhibits some form of bias. Hence, using the other 15 transformation functions is advantageous while doing comparative analyses as the other transformation functions do not cause incorrect annotation problems.

The inferences were performed on a single Nvidia RTX 3090 GPU. Most of the models were run using the HuggingFace library [86]. Hence, the models implemented in our framework have dependencies on the HuggingFace library and the reader must be aware of this fact during experimentation. Due to the expandability of the framework, multiple datasets are supported and concurrent inferences can also be performed. But for our experiments, we used a single dataset only described in the next section.

3.5.2 Dataset

Our framework was tested on the VQAv2 dataset [15] which is a standard dataset for performance evaluation of VQA models. The dataset consists of 204,721 images, 1,105,904 questions, and 11,059,040 ground truth answers in total. Due to limited computational resources, we ran our experiments on a randomly sampled subset of the VQAv2 dataset by taking 3,000 images and their corresponding 16,000 question-answer pairs and augmenting the dataset for 15 transformation functions and 5 severity levels. Disregarding the original dataset, every model was run on $15 \times 5 = 75$ augmented datasets.

To conduct our experiments, we utilized JSON files for annotations and questions. The JSON file for questions contains information such as the question ID, type, and the actual question. On the other hand, the JSON file for annotations contains answers for each question with 10 different options and confidence scores for each option. This approach provides the user with the flexibility to choose from a range of possible answers for each question, rather than being constrained to a single correct answer.

3.5.3 Models Evaluated

We have evaluated 6 different models in total using our framework. These models are: $BLIP$ and $BLIP_{large}$ [3], $ViLT$ [4], GIT and GIT_{large} [5] and finally, the zero-shot model PNP [12]. For $BLIP$ and GIT the $large$ variations have also been used to find a plausible relation between the size of the model i.e. the number of parameters and its associated robustness. $BLIP$, GIT, and $ViLT$ are state-of-the-art models with a relatively high accuracy. All three of them are transformer-based models. PNP is the only exception in our list of models; firstly, it is a zero-shot model, and secondly, it uses various modules that can be updated with time. For our experiments, the PNP model used GPT-3 [59] as the Question Answering module and BLIP [3] as the Image-Question Matching and Image Captioning module.

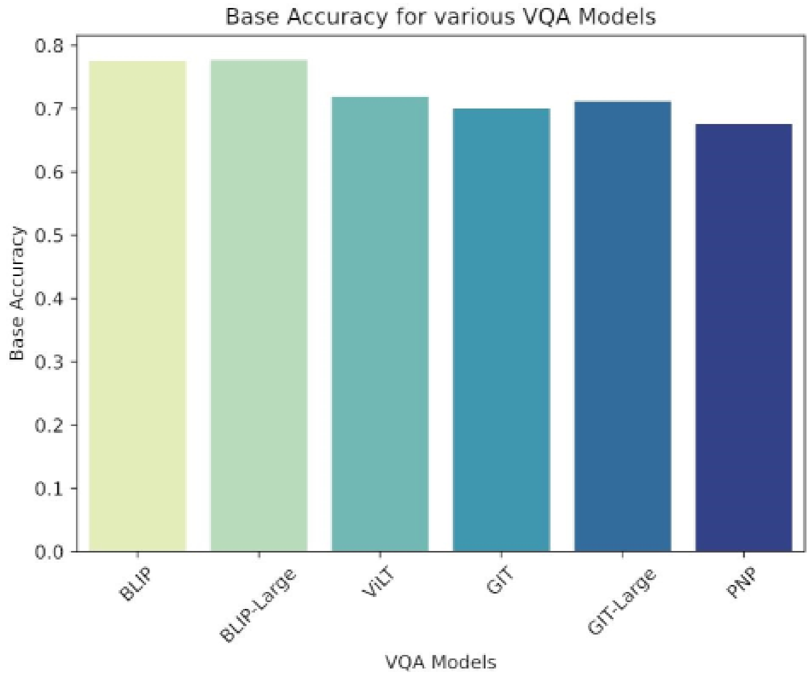


Figure 3.3: Comparison of base accuracy of the models used in this paper

3.6 Result Analysis

From the experiments, we find an error value and accuracy value for a particular model and a particular type of corruption for every severity level. Fig-3.4 helps us visualize the error for the 6 severity levels of the $BLIP$ model. Both the error and accuracy values are metric values that will later be aggregated to make more meaningful inferences. We will get many such values for various models and corruption which will be aggregated either model-wise or corruption-wise by metrics defined in the section-3.4.

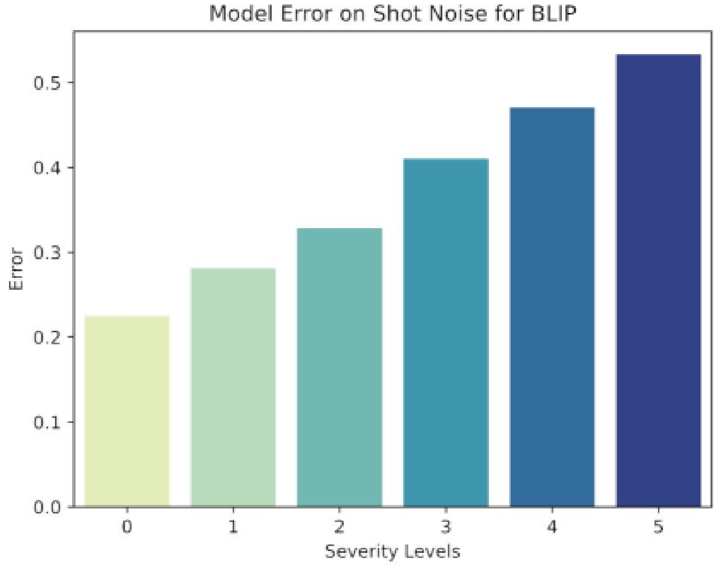


Figure 3.4: The error for every severity level for shot noise tested on the $BLIP$ model [3]. A common property of the errors values is that it increases with the severity level.

3.6.1 Comparative Analysis of Accuracy

We first have a look at the average accuracy for varying models and noise. Table-3.1 shows the results for all 6 models and for 15 categories of noise. From the table, we can infer that PNP has a comparatively lower accuracy than the other models as PNP is a zero-shot model and hence, has not been explicitly trained on VQA. The $BLIP$ models have comparatively higher accuracy than both $ViLT$ and GIT. The larger models also tend to outperform the base models. Looking at the fig-3.6, we can understand how the accuracy might vary for the various types of corruption effects. To visualize each corruption effect, the reader is advised to refer to fig-2.18 by [7] which encapsulates all the 15 corruption functions used in this paper. Firstly, the brightness function has the highest average accuracy and the outcome is not surprising as neither the model nor the humans should be confused by brighter images. On the

Table 3.1: Average Accuracy of the models for different types of corruption

Model Noise	**BLIP**	**BLIP Large**	**ViLT**	**GIT**	**GIT Large**	**PNP**
Shot	0.625	0.634	0.577	0.543	0.553	0.497
Gaussian	0.715	0.720	0.665	0.628	0.638	0.434
Impulse	0.706	0.710	0.660	0.615	0.620	0.447
Speckle	0.732	0.740	0.686	0.645	0.656	0.419
Defocus	0.706	0.710	0.673	0.656	0.665	0.425
Glass	0.702	0.709	0.652	0.619	0.637	0.458
Zoom	0.625	0.630	0.564	0.525	0.534	0.531
Snow	0.685	0.694	0.604	0.578	0.591	0.454
Brightness	0.755	0.764	0.701	0.683	0.692	0.387
Contrast	0.715	0.722	0.640	0.603	0.622	0.480
Saturation	0.729	0.734	0.681	0.644	0.653	0.319
Elastic	0.729	0.736	0.689	0.643	0.661	0.311
Pixelate	0.734	0.741	0.707	0.684	0.696	0.293
JPEG	0.738	0.743	0.706	0.655	0.668	0.294
Splatter	0.726	0.733	0.665	0.624	0.627	0.335

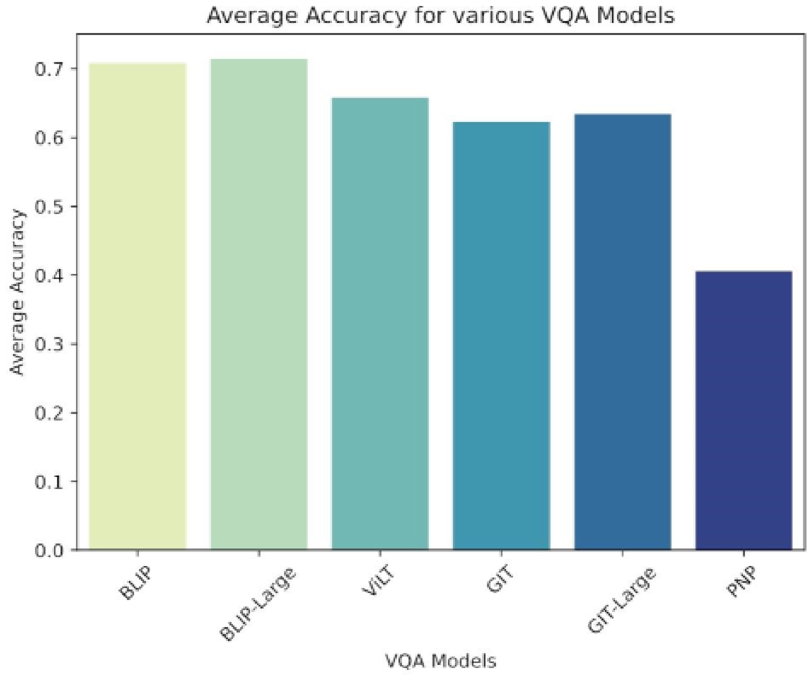


Figure 3.5: Comparison of average accuracy of the models used in this paper

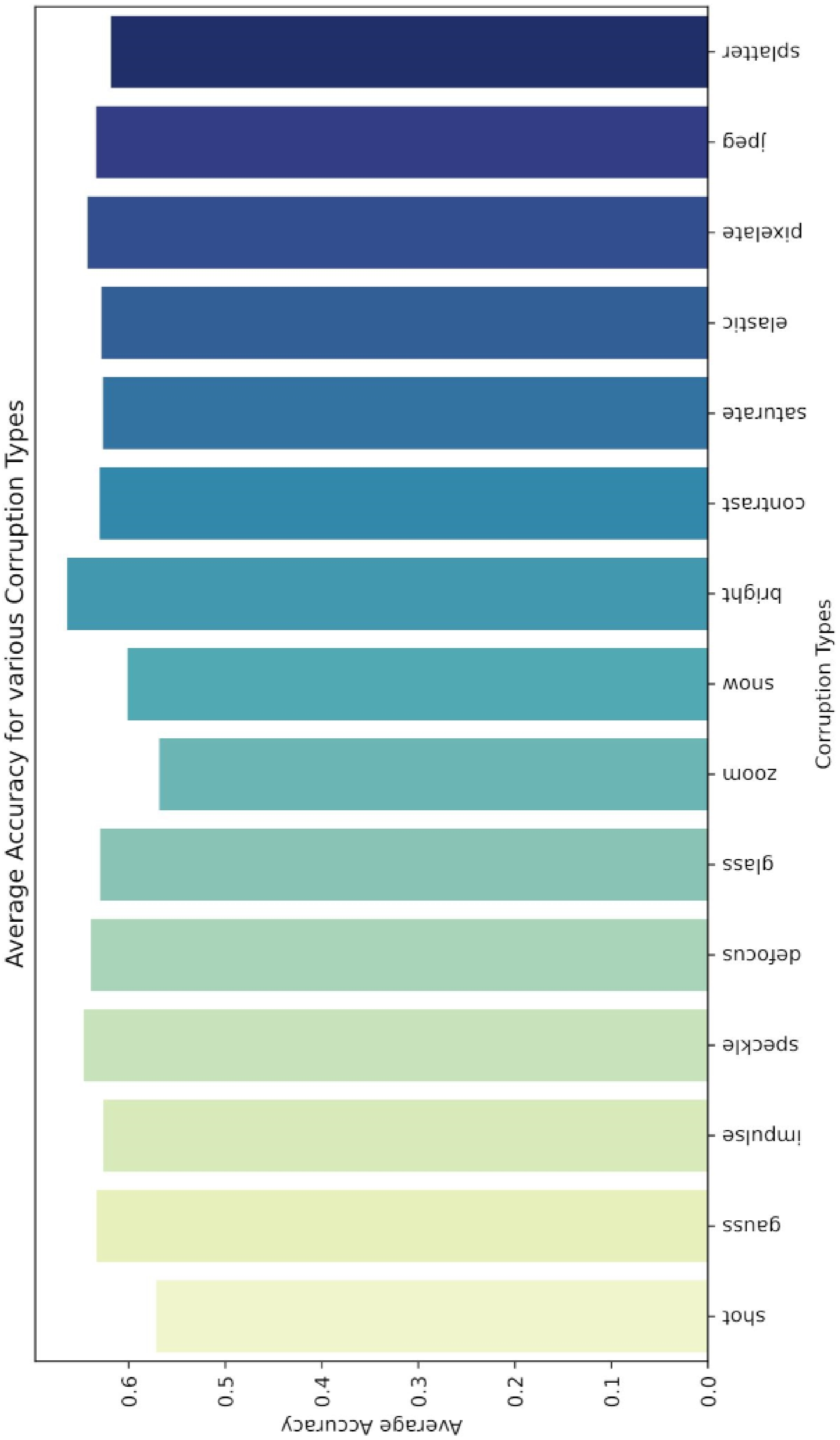


Figure 3.6: Comparison of the average accuracy for various types of corruption

other end of the spectrum, zoom blur has the lowest accuracy, and unsurprisingly, if an image appears to have a zoomed-out effect human observers tend to have a hard time answering related questions as well.

Shot noise and snow noise have slightly better accuracy than zoom blur. Both shot and snow noise have dotted particles scattered on the image which might result in blocking important image details and thus, resulting in a poor accuracy score. Splatter noise tends to have a lower accuracy as well due to similar reasons. Most of the other corruption types have similar accuracy results which doesn't give us much information to differentiate between those corruption types.

Unexpectedly, the speckle, defocus and pixelate have higher average accuracies, and looking at the corruption effects one might argue that these effects should make question-answering more difficult on the image. However, we also have to keep in consideration that there are several severity levels, and the rise of difficulty for each severity level is different for different kinds of corruption effects. Hence, we can conclude that accuracy is a simple measure of evaluating the strength of a corruption function but it may not be the best way.

3.6.2 Comparative Analysis of VRE

Table 3.2: Normalized $VRE_{v,c}$ score of the models for different types of corruption

Model Noise	BLIP	BLIP Large	ViLT	GIT	GIT Large	PnP
Shot	0.310	0.202	0.208	0.661	0.602	0.999
Gaussian	0.106	0.042	0.108	0.220	0.223	1.000
Impulse	0.138	0.121	0.095	0.629	0.577	0.931
Speckle	0.220	0.096	0.130	0.342	0.511	1.000
Defocus	0.417	0.368	0.101	0.300	0.135	0.936
Glass	0.328	0.264	0.339	0.178	0.402	0.948
Zoom	0.228	0.184	0.513	0.902	0.906	0.621
Snow	0.204	0.114	0.580	0.608	0.671	0.625
Brightness	0.137	0.000	0.162	0.482	0.550	0.969
Contrast	0.024	0.000	0.236	0.324	0.289	1.000
Saturation	0.229	0.174	0.102	0.575	0.549	0.507
Elastic	0.363	0.311	0.096	0.168	0.466	0.679
Pixelate	0.451	0.343	0.064	0.165	0.149	0.777
JPEG	0.411	0.306	0.041	0.566	0.506	0.604
Splatter	0.056	0.003	0.244	0.637	0.590	0.742

Following the limitations of using accuracy to evaluate a model's robustness, we will use our proposed metric Visual Robustness Error (VRE) in this subsection. Firstly,

Table 3.3: $VRE_{v,c}$ score of the models for different types of corruption

Model Noise	BLIP	BLIP Large	ViLT	GIT	GIT Large	PnP
Shot	0.288	0.272	0.296	0.332	0.330	0.362
Gaussian	0.184	0.178	0.203	0.225	0.223	0.352
Impulse	0.198	0.196	0.209	0.262	0.255	0.346
Speckle	0.162	0.153	0.178	0.204	0.208	0.332
Defocus	0.196	0.191	0.192	0.212	0.200	0.334
Glass	0.205	0.196	0.218	0.213	0.231	0.333
Zoom	0.308	0.302	0.327	0.351	0.352	0.308
Snow	0.227	0.215	0.271	0.274	0.279	0.321
Brightness	0.135	0.126	0.161	0.178	0.176	0.329
Contrast	0.187	0.175	0.238	0.272	0.257	1.201
Saturation	0.164	0.159	0.182	0.215	0.208	0.359
Elastic	0.168	0.161	0.178	0.201	0.210	0.372
Pixelate	0.165	0.155	0.158	0.174	0.167	0.379
JPEG	0.158	0.150	0.156	0.203	0.192	0.367
Splatter	0.165	0.158	0.197	0.250	0.243	0.371

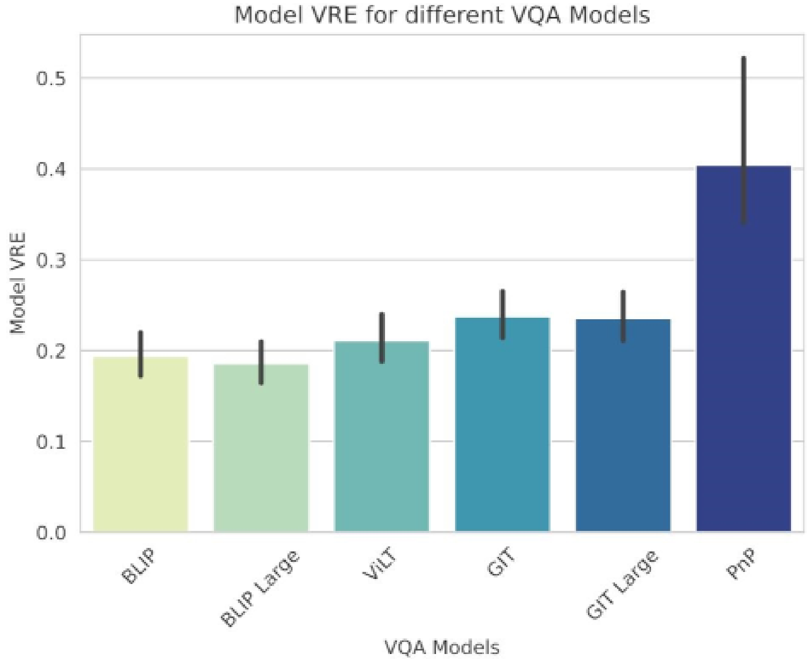


Figure 3.7: Comparison of VRE_v of the models used in this paper

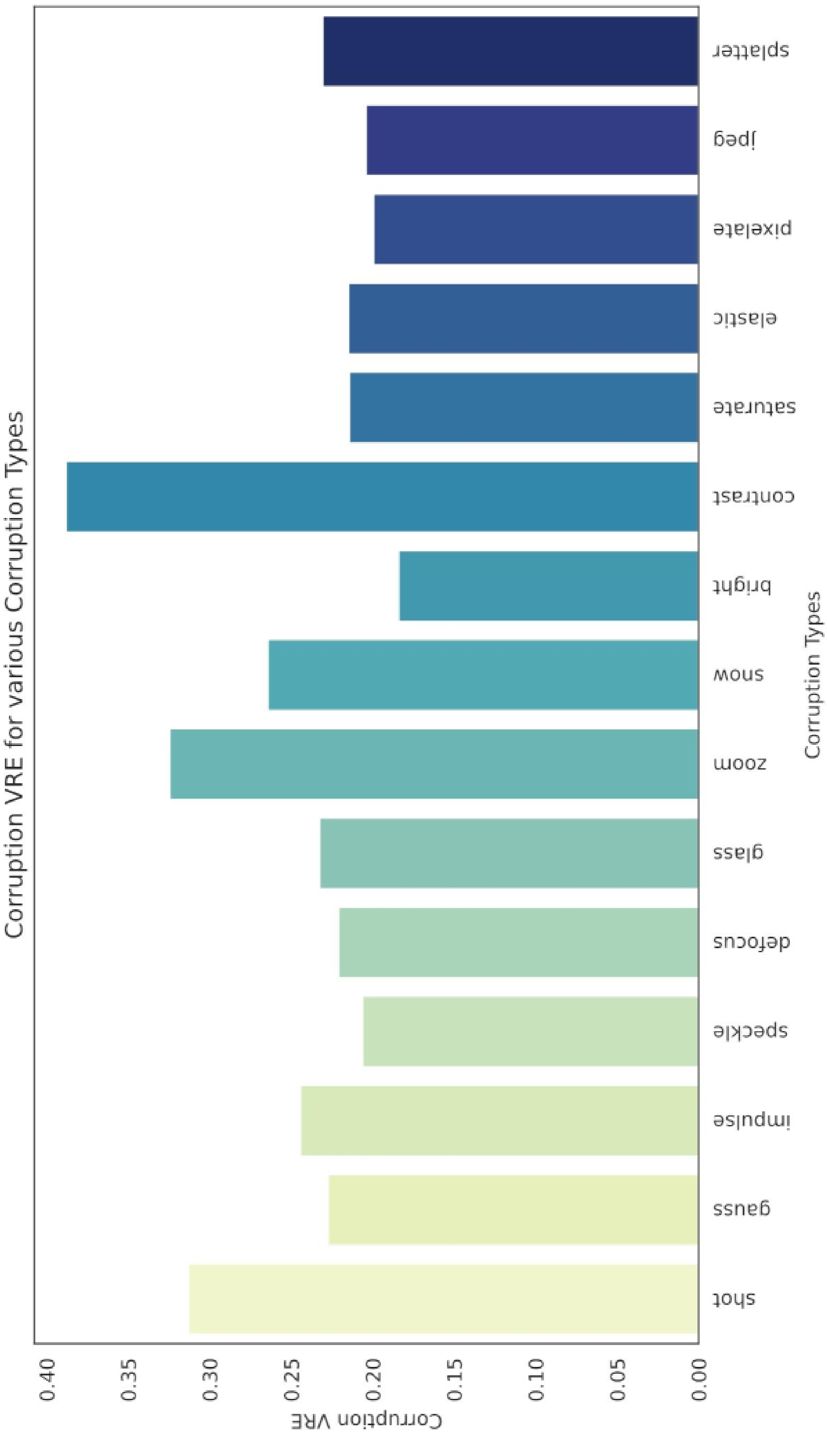


Figure 3.8: Comparison of VRE_c of various types of corruption

the metric VRE is a measurement of error; hence, the lower the VRE, the better the performance of the model. Secondly, the VRE can have different variations – VRE_v, VRE_c, and $VRE_{v,c}$ which represent VRE for a particular model v and a particular corruption c. The first two values are corruption and model aggregated values respectively while the latter one is shown in table-3.3, 3.2. All the analyses in this section are done with $\alpha = 0.5$ which is the default metric value. Hence, the value of α is not explicitly mentioned for every analysis and the effect of α will be explored in a later section.

Looking at fig-3.7, we can understand the significance of VRE_v in understanding the robustness capabilities of a particular model. In fig-3.7, the bar represents the non-normalized VRE_v score for the given models and the line represents the variation. Following the discussion from the previous section, we can now safely conclude that PNP is the least robust model, and the difference between PNP and the second-least robust model is clear in the figure. PNP matches most of the criteria of a model with lower robustness which is evident from VRE_v.

The $BLIP$-based models tend to be the most robust models close to $ViLT$ and GIT-based models. So far, the VRE results are consistent with the average accuracy scores and hence, none of our models severely lack robustness compared to the average accuracy. As the model sample state is small, no noticeable difference between the two metrics is seen. However, the key difference between VRE and average accuracy is seen while analyzing the 15 corruption functions.

The contrast effect has the highest VRE value making it the strongest corruption function while the contrast value has a significantly high average accuracy. When images with high contrast are passed, the models tend to perform poorly at lower severity levels. However, the error values are always relatively low. Hence, averaging them will give a relatively low error value but when taking other metrics like First-Drop into consideration, the contrast seems like a *stronger* corruption.

Brightness is still the weakest corruption type with the least VRE_c value. Shot noise and zoom blur also have higher VRE_c values aligning with the results from average accuracy. One thing the reader can infer by comparing fig-3.6 and fig-3.8 is that the latter figure has more variation and thus it is easy to differentiate between the bars. The average accuracy is more or less similar and a user of our framework might misunderstand this as an insignificance difference between the types of corruption. But in reality, average accuracy isn't a good metric to highlight the differences.

3.6.3 Model Size and Robustness

Through our rigorous experimentation, we discovered that increasing the size of a model does not necessarily translate into enhanced robustness. While there is a marginal improvement in accuracy as we increase the model size, the gains in robustness are rather modest. This finding suggests that simply scaling up the size of a model does not guarantee a substantial boost in its ability to handle diverse and challenging inputs. Although larger models tend to possess a greater number of parameters and a higher level of representational capacity, this alone does not result in significant improvements in robustness. It highlights the importance of other factors such as the training process, the diversity and quality of the dataset, and the architectural design in determining the model's ability to handle variations and uncertainties.

Moreover, the slight increase in robustness observed with larger models suggests that there might be diminishing returns beyond a certain model size. As the model grows larger, the marginal improvements in robustness become progressively smaller, while the associated computational and memory costs continue to increase. This insight prompts us to consider alternative strategies, such as architectural modifications or specialized training techniques, to achieve substantial gains in robustness without excessively inflating the model size. Fig-3.9 shows the comparison between the error of two of our models and the difference is negligible.

Additionally, it is worth noting that while the accuracy of the model sees a slight improvement with increased size, the trade-off between accuracy and robustness remains evident and this topic will be discussed in a later section. Sometimes, as the model becomes more accurate, it may become more sensitive to variations or perturbations, potentially compromising its overall robustness. This reinforces the need for a balanced approach that carefully weighs the desired level of accuracy against the requirement for robustness in specific applications.

To summarize, our experiments have revealed that the relationship between model size, accuracy, and robustness is complex. Merely increasing the size of a model does not guarantee significant gains in robustness, although there is a slight improvement observed. This finding calls for a more nuanced approach to model development, where other factors and techniques are considered alongside model size to enhance both accuracy and robustness. Future research should explore alternative strategies that optimize both aspects effectively, leading to more reliable and versatile machine learning models.

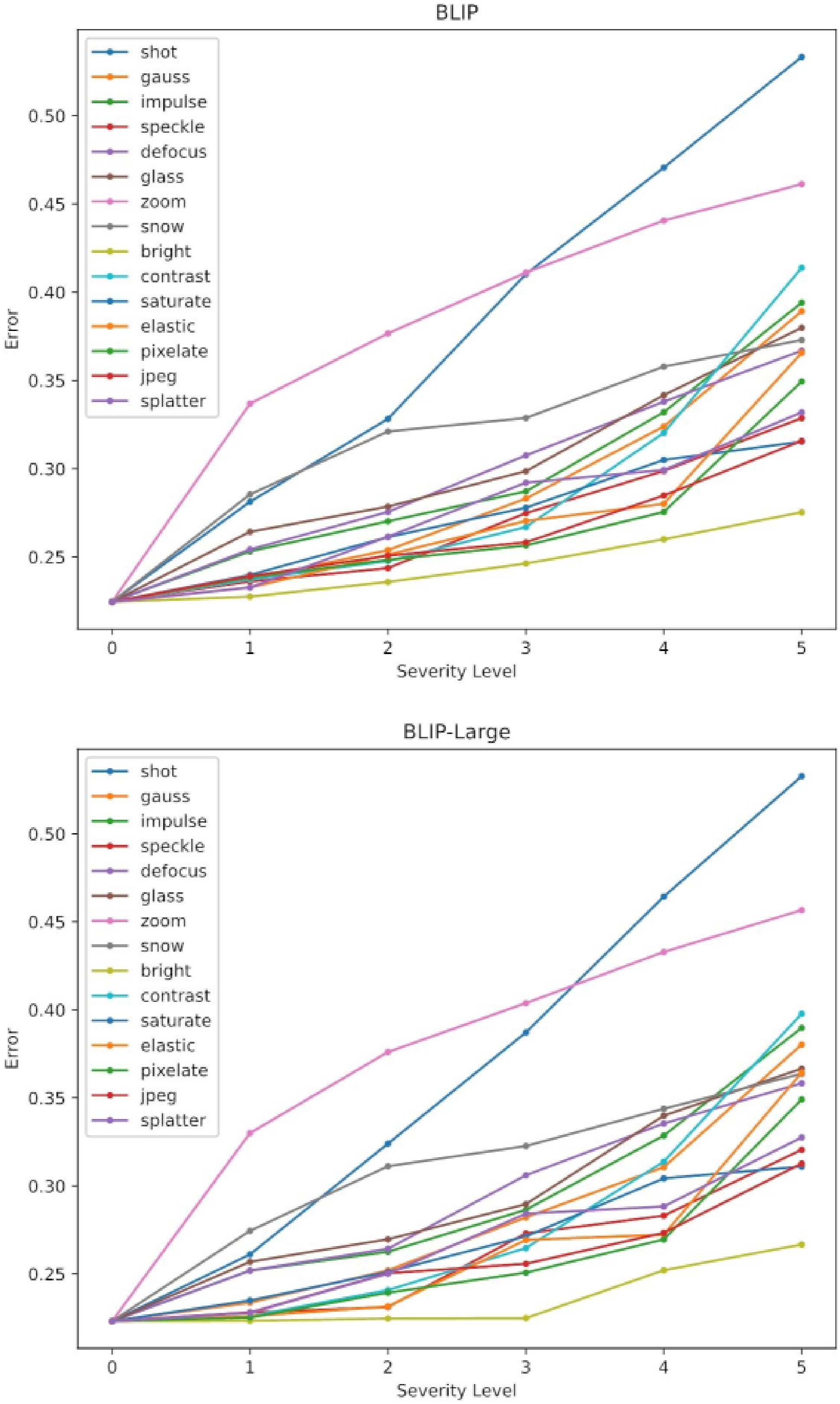


Figure 3.9: Comparison of error values for BLIP and BLIP-large [3]

3.6.4 Question Type and Error

The availability of question-specific data enables us to inform users about the limitations of the models in providing accurate responses to different question types, along with the corresponding error rates. This information serves to highlight the model's weaknesses, particularly when subjected to the introduction of noise in the form of additional images. By disclosing the types of questions that the model may struggle to answer and quantifying the associated error rates, we can effectively demonstrate the areas where the model may exhibit limitations. This insight contributes to a comprehensive understanding of the model's weaknesses and its implications in practical applications.

Table 3.4: Effect of Noise on ViLT [4] Predictions

Question Type	Questions	Answers	Predictions
Color	What color is the lamp?	blue	white
	What color is the bike?	blue	black
	What color is the sky?	blue	gray
	What color is the soap on the wall?	yellow	white
Counting	How many spoons are there?	2	1
	How many people are in the picture?	5	3
	How many kites are up?	4	3
Classification	What is she eating?	sandwich	cake
	What is the weather like?	sunny	cloudy
	What is on display?	toys	vases
	What game is this?	baseball	soccer
Blindness	What's on the television?	baby	nothing
	Are the women selling something?	yes	no
	What is the cat eating?	cake	nothing
Logical Reasoning	Which bowl has more oranges?	front	right
	What kind of car is in the picture?	Ferrari	red
	What is the man at the top about to do?	run	bat
	What is the man doing?	standing	flying kite

From table-3.4 analysis of ViLT's [4] performance in the presence of noise allows us to draw conclusions regarding the robustness of the model. It is essential to acknowledge that expecting ViLT to provide accurate answers at Level 5, similar to those at Level 0, would be unreasonable. However, the impact of noise on the model's accuracy does not show a significant decline across different levels, suggesting a noteworthy level of robustness. Furthermore, it is worth noting that images with Level 5 intensity are rare occurrences, indicating that ViLT performs admirably even when exposed to noise, underscoring its importance as a valuable model.

Moving forward, we intend to explore the performance of ViLT on various datasets to gain insights into its capabilities. We have chosen to demonstrate ViLT's performance on specific categories within the VQAV2 dataset to understand its predictive abilities in those areas. To begin, we focus on questions related to color. Within the dataset, ViLT made approximately 2762 mispredictions, with 201 of them being incorrect predictions specifically related to color. Out of the 10,000 questions, 1695 pertain to color, showcasing an 88.14% accuracy rate in detecting colors. This suggests that ViLT performs remarkably well in accurately identifying colors, demonstrating that the "color" category does not significantly contribute to misclassifications.

Next, we examine the counting questions where ViLT made around 2762 mispredictions, with 449 incorrect predictions for such questions. Out of the 10,000 questions, 1771 fall into this category, resulting in a 74.64% accuracy rate in counting data. While it may not be considered highly reliable, ViLT can still be utilized regularly for counting tasks. It is important to note that the 25.12% inaccuracy includes predictions that are reasonably close to the actual answers, and logical reasoning can often be employed to address such challenges, which can prove demanding for any model.

Furthermore, we present examples that demonstrate ViLT's poor performance in logical reasoning tasks. While these inaccuracies do not account for a significant portion of the overall errors, it is worth mentioning that ViLT occasionally produces results that could be considered as correct in some cases. The provided table showcases examples falling into this category.

Lastly, in order to evaluate the model's proficiency in text recognition, we specifically focus on questions containing the word "say" as it frequently appears in textual content accompanying images. Out of the 2762 mispredictions made by ViLT, 44 of them involve incorrect predictions related to "say" questions. Among the 10,000 questions, 105 pertain to "say," resulting in a 58.09% accuracy rate in predicting texts. This indicates that ViLT may not be the most suitable choice for tasks involving text extraction from images.

ViLT's robustness in the face of noise and its impressive performance in accurately identifying colors. While it may exhibit limitations in logical reasoning and text extraction, ViLT still delivers commendable results and can be effectively utilized in various applications.

3.7 Metric Analysis

In this section, we explore the effect of our proposed metric VRE in-depth and analyze the effect of its constituting metrics. We further explore the parameter α in our metric and how it varies depending on the robustness use case.

3.7.1 Effect of Normalization of Metric Scores

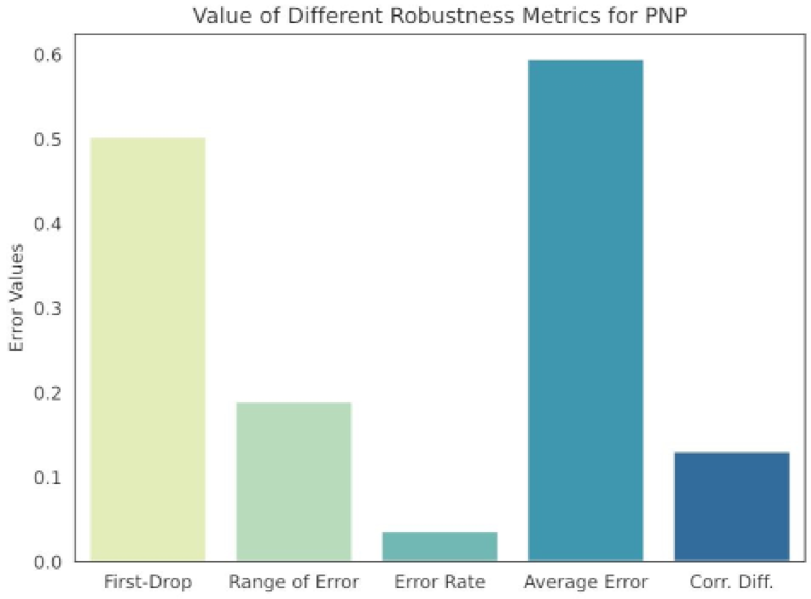


Figure 3.10: Comparison of metric values for PNP [12]

Fig-3.10 explores how every metric contributes to the VRE_v of PNP. Typically, the error rate has lower values, and hence, for PNP, the average error is the most important factor for VRE_v. If we also take into account that due to $\alpha = 0.5$, the average error will have four times the weight of every other metric and would hence make VRE_v work similar to A_v. To prevent this, normalization is performed to have similar values for every metric.

Fig-3.11 performs model-based normalization according to equation-3.13 and has most of the metric values in a similar range. PNP still has a high average error but the other metrics are not as insignificant. The same rationale applies for every other model and we can attest to the necessity of normalization of the metric values. Non-normalized values can also be used but the weights assigned to every metric need to be manually set. Such practices are discouraged due to the ambiguity related to establishing a linear relationship between the metrics while aggregating them.

3.7.2 Composition of Metrics in VRE

Table-3.5 shows the scores of the proposed metrics when aggregated into VRE. The scores are non-normalized and similar to fig-3.10, we can see some of the metrics have comparatively lower values. However, it is unclear which model excels in which metric. In ideal scenarios, we can assign weight to each metric during aggregation. Hence, it is essential to understand the effect of each metric in establishing a VRE

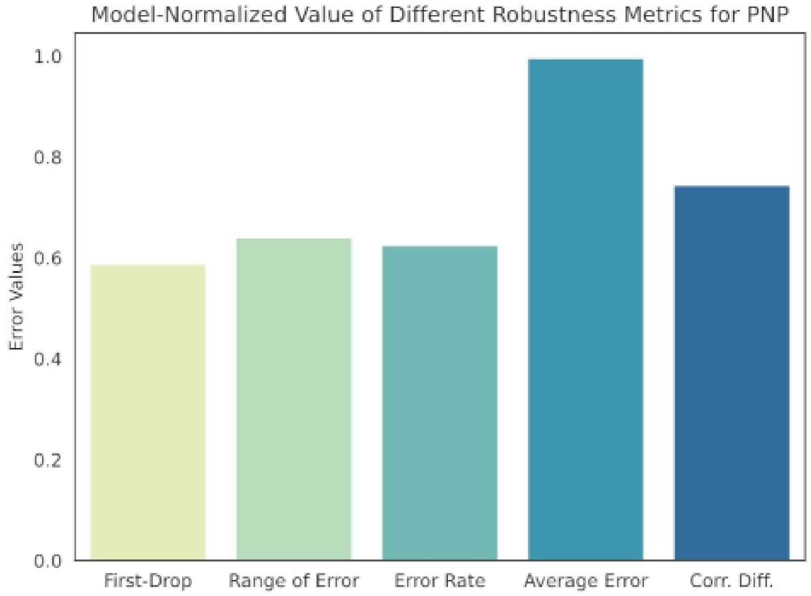


Figure 3.11: Comparison of normalized metric values for PNP [12]

Table 3.5: Non-normalized metric values of the models for different metrics

Metric / Model	First Drop	Error Range	Error Rate	Avg. Error	Corr. Dif.
BLIP	0.619	0.428	0.403	0.025	0.382
BLIP-Large	0.359	0.358	0.369	0.001	0.262
ViLT	0.2	0.165	0.173	0.226	0.167
GIT	0.524	0.58	0.545	0.361	0.512
GIT-Large	0.638	0.662	0.614	0.319	0.611
PnP	0.587	0.64	0.625	0.996	0.744

value that is suitable for real-life applications.

Fig-3.12 gives us an illustration of how the normalized scores for all 6 models contribute to their VRE. For ViLT [4], all the scores are relatively low and hence, making the model extremely robust as well as accurate. But, our most robust model BLIP [3] has the least average error but the other metrics are relatively higher than ViLT. Hence, we can conclude that if our application disregards or underappreciates the model's accuracy, then ViLT is the most suitable model. Models like PNP [12] and highly inaccurate with high robustness error, hence, indicating that PNP is an unfavorable model in any scenario. In the next section, we dive deeper into the tradeoff between accuracy and the overall robustness of the model.

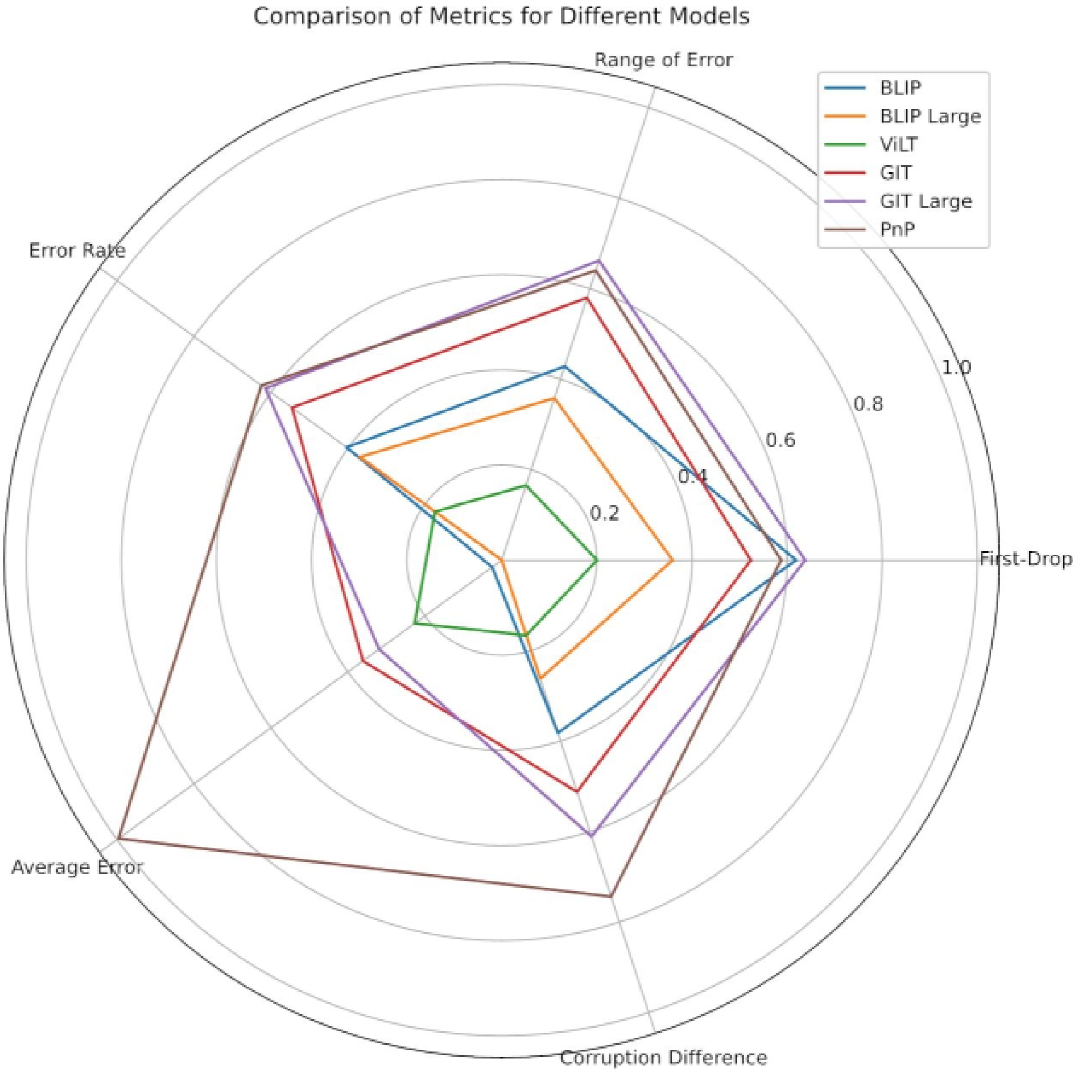


Figure 3.12: Composition of Normalized Metric values in VRE for various models

3.7.3 Accuracy vs Robustness: Effect of α

Through our extensive experimentation, we have made a significant observation that sheds light on an important trade-off in model development: the delicate balance between accuracy and robustness. We can use a simple analogy to illustrate this fact; suppose, we have two models – a severely underfit model that always gives that has 10% accuracy and a better model with 80% accuracy. Based on the accuracy scores, is it possible to infer which model is more robust? On the contrary, if we apply a small amount of noise to the input image, then the first model's accuracy is retained at 9% while the second model's accuracy drops to 55%. We can now infer which model is more robust. This finding underscores the notion that achieving high accuracy in a model often comes at the expense of its robustness, and vice versa.

When a model is optimized for accuracy, it tends to excel in providing precise

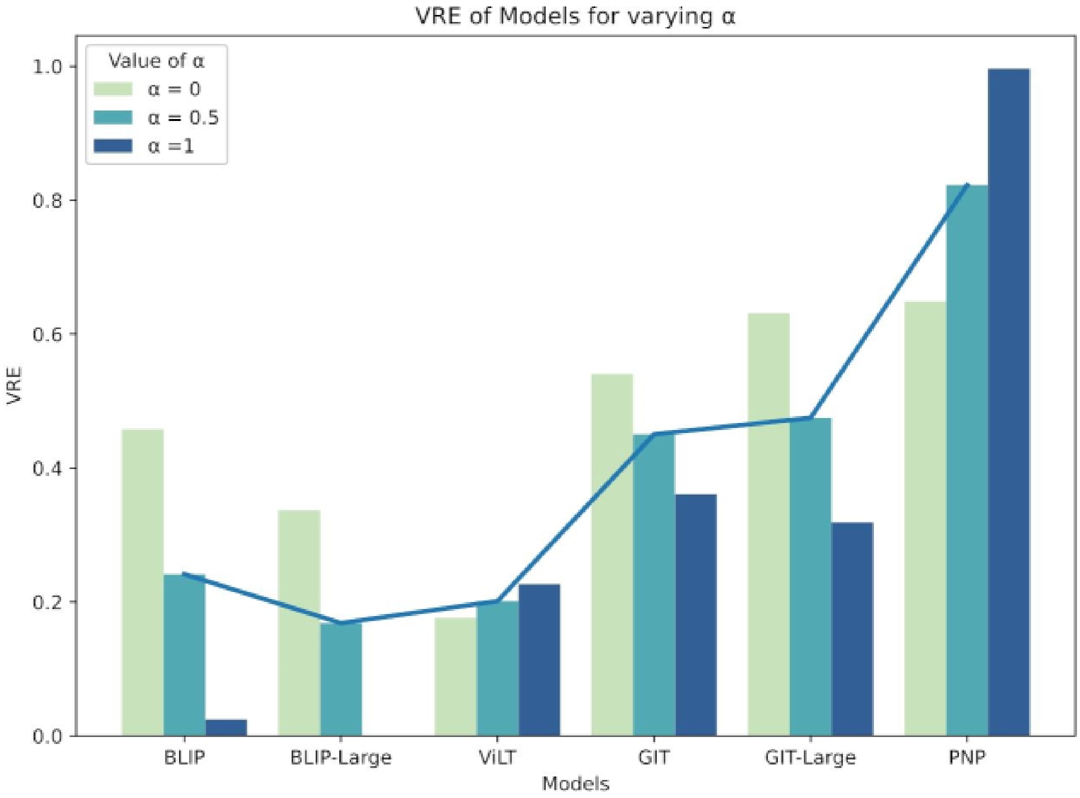


Figure 3.13: The change of VRE for different models with varying values of α

and correct predictions under ideal conditions. However, it becomes more susceptible to faltering or producing erroneous outputs when faced with variations, uncertainties, or adversarial inputs. On the other hand, a robust model exhibits a higher level of resilience and generalization, capable of performing consistently across a wide range of inputs, even in the face of perturbations or challenging scenarios. However, this increased robustness might come at the cost of sacrificing some accuracy, as the model adopts a more conservative or cautious approach to minimize errors.

The trade-off between accuracy and robustness is crucial to consider when developing machine learning models for various applications. Different contexts and use cases may require varying degrees of emphasis on accuracy and robustness. For instance, in safety-critical systems, such as autonomous vehicles or medical diagnosis, robustness takes precedence over accuracy to ensure reliable performance even in uncertain or unpredictable situations. On the other hand, in tasks where precision and correctness are paramount, sacrificing some robustness may be acceptable to achieve higher accuracy.

Understanding this trade-off enables researchers and practitioners to make informed decisions when designing models, striking a balance that aligns with the specific requirements and priorities of the given application. It also highlights the need for com-

Table 3.6: VRE score of the models for values of α

Model Noise	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
BLIP	0.619	0.428	0.403	0.025	0.382
BLIP-Large	0.359	0.358	0.369	0.001	0.262
ViLT	0.2	0.165	0.173	0.226	0.167
GIT	0.524	0.58	0.545	0.361	0.512
GIT-Large	0.638	0.662	0.614	0.319	0.611
PNP	0.587	0.64	0.625	0.996	0.744

prehensive evaluation metrics that consider both accuracy and robustness, providing a more holistic assessment of model performance. Fig-3.13 illustrates how the VRE is affected at various levels of prioritization of accuracy and robustness. Our findings emphasize the delicate interplay between accuracy and robustness in machine learning models. Recognizing and managing this trade-off is essential for developing models that align with the desired performance objectives in various real-world scenarios.

3.8 Discussion

3.8.1 Mislabeling Problem in Grayscale Images

Grayscale images are void of color and hence, questions related to color shouldn't be answerable by the model inferring a grayscale image. Table-3.7 shows some color-related questions that cannot be answered from a grayscale image and if answered, it will be a shade of grey. However, most of our models picked a particular color as the answer to the image indicating that the model associated a color to a particular shape or structure in the image. For instance - if the model sees the gray image of an apple, and is asked "What is the color of the apple?", it will be tempted to predict "Red" as most of the images of apples it was trained on had the color red. Hence, it associated the color red with the shape of the apple.

As we compare the answers to the ground truth answers in our dataset, we realize that the color answers predicted by the model are given full scores which is incorrect as a gray image is void of color. Hence, for color-related questions, relabeling needs to be done or we will end up with a mislabeling problem that will inaccurately assess a model's robustness. As we did not perform any form of relabeling for grayscale images and inverted grayscale images, we do not include these forms of corruption while calculating VRE. However, our framework includes these two corruption effects for other forms of analysis.

Table 3.7: Color-based Questions that might cause mislabeling problems

Questions	**Answers**
What color is the majority of the fruit on the table?	yellow
What color is the water?	blue
What is the red-colored topping on the pizza?	sauce
What color is the coca/coca light?	red
What color is the lettuce?	green
What color is the tree on the right?	green
What is the dominant color of this road sign?	red

3.8.2 Robustness of PNP

Looking at the PNP architecture [12], we can highlight two significant points about the model. Firstly, the architecture is modular, and hence, the overall robustness of the model will depend on the individual robustness of each module. We can also formulate as:

$$VRE_{PNP} = 1 - \prod_{m}(1 - VRE_m) \tag{3.19}$$

where m is a module of PNP. As VRE in both normalized and non-normalized form is lesser or equal to 1, $(1 - VRE_m)$ which can also be termed as *robustness accuracy* is also lesser or equal to 1. Hence, the product of many such values can never increase and hence, we can conclude that for zero-shot-based models with multiple modules, the VRE_v tends to increase with the number of modules given that each module has a similar level of robustness.

In reality, every module has a different purpose and should have different levels of robustness. There can be various types of modules - image-question matching module, image captioning module, question-answering module, etc. Trivially, we can say the question-answering module is unaffected by any visual noise and can hence, be excluded from the list of modules. An image-question matching module and an image captioning module are both multimodal modules but with different architectures. However, passing noise-free images to any one of the two modules, we found that there isn't a statistically significant difference between the VRE scores of the two modules.

Some of the earlier approaches to Zero-Shot VQA explored models [35] that use feature extraction, and representation similarities. These architectures are not modular and have poor zero-shot accuracy. Modern architectures are Large Language Model (LLM)-based and tend to be more robust than traditional non-modular architectures due to being pre-trained on substantially higher training data.

3.8.3 Zero-shot and Robustness

In our comprehensive series of tests, we have made a compelling observation that sheds light on the performance characteristics of zero-shot models. Specifically, we have found that these models exhibit remarkable robustness, surpassing other models in terms of their ability to handle diverse and challenging inputs. However, this robustness comes at the expense of accuracy, as zero-shot models tend to yield significantly lower scores in this aspect.

The high level of robustness displayed by zero-shot models is a noteworthy finding. It implies that these models have a greater capability to generalize and handle unseen or out-of-distribution data. They are adept at adapting to different contexts and inputs, making them more resilient to variations and uncertainties. This characteristic is particularly valuable in real-world applications where encountering new or unexpected scenarios is common.

On the other hand, the trade-off between robustness and accuracy becomes apparent with zero-shot models. While they excel in robustness, their performance in terms of accuracy is comparatively lower. This suggests that zero-shot models may struggle with fine-grained or nuanced tasks that require precise and detailed predictions. Their focus on generalization and adaptability may result in broader but less precise outputs, leading to a decrease in accuracy scores.

These findings underscore the importance of striking a balance between robustness and accuracy in model development. The choice between a zero-shot model and other models depends on the specific requirements of the task at hand. If robustness and adaptability to diverse inputs are crucial, a zero-shot model may be the preferred option. However, for tasks that demand higher levels of accuracy and precision, alternative models that prioritize accuracy might be more suitable.

It is worth noting that these results also highlight the need for further research and development in improving the accuracy of zero-shot models. Finding ways to enhance their performance in terms of accuracy without compromising their robustness could unlock their full potential and make them more viable for a wider range of applications.

In summary, our tests have shown that zero-shot models excel in robustness while exhibiting lower scores in accuracy. This trade-off highlights the need to carefully consider the specific requirements of the task when selecting a model. Future research should focus on improving the accuracy of zero-shot models to make them more competitive in domains that demand high precision and fine-grained predictions.

Chapter 4

Zero-shot VQA

In this chapter, we perform a comprehensive study on the current approaches of Zero-Shot VQA, and the shortcomings of the current models, followed by our proposed methodology. Then we explore how our proposed methodology improves upon previous architectures by performing various comparative analyses and visualizing them. We end the chapter by discussing the limitations of our model and possible approaches to solving them.

4.1 Background Study

In the previous chapters, we have seen how VQA evolved to Zero-Shot approaches. Zero-shot VQA uses pre-trained models that are not specifically trained on VQA. With the rise of computational capabilities, the pre-trained models started getting larger bringing us to the era of Large Language Models (LLMs) [24, 68, 70] and models that combine image and text modalities [3, 4]. Leveraging LLMs helped zero-shot VQA architectures achieve monumental results along with certain shortcomings.

Using LLMs as building blocks for larger architectures, Zero-shot VQA networks began a modular approach for network architectural design [12]. Looking at figure-2.25, we see multiple modules interacting with each other using attention and image captions. The image caption can be seen as an interface that links the captioning module with the question-answering module. The better the interface, the stronger the relationship between the two modules, hence, generating a better answer.

The PnP architecture [12] generates 100 captions and uses the whole set of captions as the interface. The set contains captions of varying quality and the whole set is processed by the question-answering module, usually an LLM. Using such a high number of captions has two major drawbacks - firstly, a higher number of captions indicates the presence of a high number of low-quality captions which essentially acts as noise to the question-answering module. Secondly, LLMs have a high inference time,

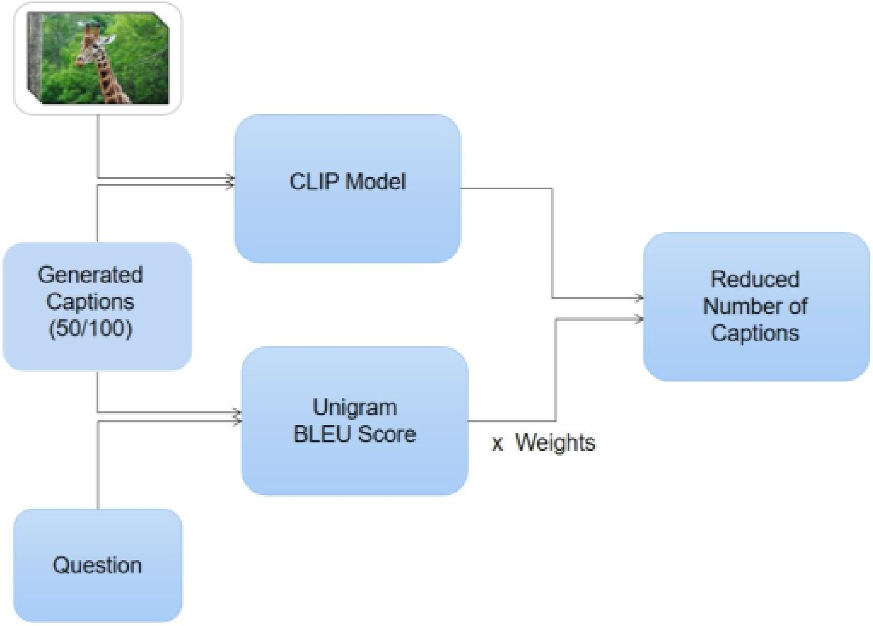


Figure 4.1: Caption Ranking Module using CLIP [9] and BLEU score [20].

and hence, the overall inference time of the architecture will be very high. Instead of having a large number of captions of varying quality, having fewer but high-quality captions should solve these drawbacks.

4.2 Proposed Methodology

Our architecture fundamentally tries to connect a pre-trained language model and a pre-trained vision-language model without any specific training on VQA and using natural language captions as the interface. Attention is used to focus on the desired part of the image based on the question. The modular approach of our architecture is similar [12]. However, we incorporate some form of caption ranking in our architecture.

While designing the caption ranking module, we kept in mind that the captions should cover the information in the image that is relevant to the question. To ensure image-caption relevance, we have a similarity module that takes the question-image pair as the input to give us a similarity score. In our architecture as seen in figure-4.1, we used CLIP [9] as the image-question relevancy module. For the caption-question relevancy, we used a well-established evaluation metric called Unigram Bilingual Evaluation Understudy (BLEU) score [20] which evaluates text quality with respect to another text. Traditionally, the BLEU score has been used to evaluate machine translation but, in this paper, it is used as a text-text relevance metric.

The interaction between the modules of our architecture can be summarized:

1. Firstly, the image-question pair is given to the image-question matching module

that uses attention to focus on relevant parts of the image.

2. The output will be sampled and passed on to the image captioning module that produces a set of captions.

3. The set of captions is then ranked based on question relevancy using the caption ranking module. The caption ranking module takes the image and the generated captions as input to produce a reduced set of captions.

4. Finally, a question-answering module takes the reduced set of captions and the question as inputs to generate the answer as the output.

In the following subsection, we will be delving deeper into the architecture of the caption-ranking module.

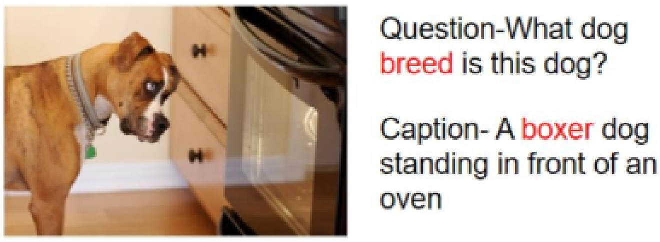
4.2.1 Caption Generation

Following the modular approach of PnP [12], we use two pre-trained modules to generate a set of captions that will be later passed to the ranking module. Referring to figure-2.24, we adopt BLIP [3] as the image-question matching module and a detailed explanation on BLIP has been discussed in section-2.4.3. The output will be later passed on the GradCAM [19] to focus on the relevant parts of the image.

The interpretation of the image by the GradCAM is encapsulated by equation-2.2 and 2.3 which can be described as the calculation of cross-attention values on the input, followed by producing partial derivatives with respect to the cross-attention values. The GradCAM output is sampled into K patches and the patches are passed on to the image captioning module. Following the authors of PnP, we also used BLIP as the captioning module which generates a set of unranked captions. The number of generated captions can be considered as a hyperparameter but following the results from [12], we generated 100 captions that are later passed to the ranking module.

4.2.2 Relevancy in Captions

Figure-4.2 is a perfect example of how we can generate something that completely resembles the question itself. As the question was searching for the breed of the dog, the caption encapsulates the idea by summarizing the whole image. We established that the captioning module works by taking attention-masked images and generating captions for the question-answering module. Our ranking module will reduce the number of generated captions by first ranking all the generated captions, and then passing the top n captions to the question-answering module.



Question-What dog breed is this dog?

Caption- A boxer dog standing in front of an oven

Figure 4.2: Generating captions that are relevant to the question.

Let, $c \in \mathbb{C}$ be a caption from our set of captions, such that, $r(c)$ would produce a score for the generated caption where r is the ranking function. For an image-question pair (i, q), we get the image-caption relevancy ranking $r_{i,c}$ = CLIP(i, c) and $r_{q,c}$ = BLEU(q, c). We can, hence, define the ranking function as:

$$r(c|i, q) = (1 - \beta)\text{CLIP}(i, c) + \beta\text{BLEU}(q, c) \tag{4.1}$$

where β is the weight assigned to the unigram BLEU score.

The idea behind ranking the captions is to pick the top n captions and thereby, generate a set of high-quality captions only. To caption is deemed as high-quality only when the caption is relevant to the image-question pair. Both CLIP and BLEU incorporate some form of relevancy measure, and hence using a weighted average of the two relevancy scores gave us a caption ranking score.

4.2.3 Image-Caption Relevancy

Image and question belong to different modalities and there's ongoing research on combining the modalities [3, 4] while CLIP [9] is one of them. Following our previous discussion, the CLIP module used for image-question relevancy gives us a similarity score between the question and images. Hence, by using CLIP, we gain insights into the semantic relationships between images and captions.

CLIP captures image-text similarity by training a transformer-based encoder for both images and captions. However, instead of having two separate modalities for the image and caption encodings, CLIP tries to minimize the dissimilarity between the two vectors by simply using scaled cosine similarity as the loss function. Hence, after the pre-training phase, CLIP should have the same or similar vector representations of an image and its caption. We can now define the contrastive loss of CLIP as:

$$\mathcal{L}(i, c) = e^{t}[N(E_{img}(i)) \cdot N(E_{caption}(c))] \tag{4.2}$$

where $\mathcal{L}(i, c)$ is the loss function for the image-caption pair (i, c), $N(\cdot)$ is the normalization function, E_{img} and $E_{caption}$ are the image and caption encoders respectively, t is the scaling constant. The loss function is also called scaled cosine similarity. Another benefit of using CLIP is its modularity i.e. any image and text encoder can be used in CLIP.

4.2.4 Question-Caption Relevancy

The task of question-caption relevancy is easier as both question and caption belong to the same modality. Hence, we can now use traditional unimodal approaches to find the similarity between two textual encodings. The easiest and most popular way to find the similarity between two vectors is the cosine similarity defined as:

$$cosine(A, B) = \frac{A \cdot B}{|A||B|} \tag{4.3}$$

Most of the other relevancy approaches build on the cosine similarity function. However, the simplicity of cosine similarity has its own drawbacks and led to the development of more nuanced metrics. Unigram BLEU score is defined as:

$$\text{Unigram BLEU} = \text{BP} \times \exp \quad \sum_{n=1}^{N} w_n \log p_n \Bigg) \tag{4.4}$$

where BP is the brevity penalty which compares the length of the two texts. The n represents the maximum n-gram order and in our case, it is unigram; hence, $n = 1$. The n-gram model calculates the probability of the occurrence of certain terminology, given n such terminologies have already occurred. The p_n and w_n are the precision of the n-gram and the associated weight respectively.

4.2.5 Aggregated Ranking Score

We combine the CLIP and BLEU score that represents the image-caption and question-caption relevancy respectively by performing the weighted average of the two scores where the weight to the question-caption relevancy score is denoted by β. Firstly, our ranking must contain both visual and textual relevancy with the generated captions. But we do not know the relationship of the two metrics in the overall ranking of the generated captions. An assumption of a linear relationship has been made and the associated weight in the linear relationship is treated as a hyperparameter.

The mathematical formulation of our aggregated ranking is observed in equation-4.1. Experimental results showed $\beta = 0.6$ as the optimal value to maximize accuracy for the CLIP+BLEU method while keeping the same inference time. β is a tuneable hyperparameter that will be based on the relevancy methods used. The aggregated ranking score also reduced the number of captions to just 5 and thereby providing a vast improvement to inference time with minimal overhead.

4.2.6 Answer Generation

The reduced set of captions is passed to the question-answering module to generate the final output which is the answer itself. For now, we will be working with GPT-3 [59] as it gives out answers on text with the passage and gives questions. Since the captions contain what is needed, the question-answering module will be going through those and get the necessary information out and show the result.

4.3 Performance Evaluation

In this section, we go through our setup, the model used, and the dataset for inference. We conclude the section by looking at our evaluation metric and defining any proposed metric if necessary.

4.3.1 Experimental Setup

As the setting of the experiments is zero-shot, pre-trained models were used. Hence, no setup had to be done for training. Experiments are primarily inference-based and the inferences were done on a single Nvidia RTX3090 GPU. We used Jupyter Notebooks to run our experiments in the Python programming language.

4.3.2 Model and Dataset

The evaluated models are a set of zero-shot models that are derivations of PNP [12]. The derivations include the addition of various ranking modules to our model. Primarily, cosine similarity, BLEU score [20], and CLIP [9] are used as sub-modules inside the ranking module.

The dataset is the standard VQAv2 dataset [15] validation split. No form of preprocessing has been done for benchmarking purposes. A detailed description of this dataset split can be found in section-3.5.2.

4.3.3 Evaluation Metric

In our work, we use three evaluation metrics – accuracy, inference time, and relative inference difference. The accuracy is simply the misclassification accuracy for VQA as defined in equation-3.3. The inference time, $t_{V,R}$, can be defined as the time required to generate the answer for a particular model V using a ranking approach R. For a baseline model V_0 and baseline ranking method R_0, we can hence define the relative inference difference which calculates the *speedup* of the model as:

$$\Delta t_{V,R} = \frac{t_{V_0,R_0} - t_{V,R}}{t_{V_0,R_0}} \tag{4.5}$$

In our work, we use the PnP model [12] with 100 generated captions as the baseline model and *no ranking* as the baseline ranking method.

4.4 Result Analysis

The results obtained in our study focus on the accuracy of different methods and their inference time, considering the number of captions and the weight of the Unigram BLEU score. Upon analyzing the results, it becomes evident that we can achieve good accuracy with fewer captions. In fact, using only 5 captions yields significantly better results compared to using 50 or 100 captions.

Table 4.1: Comparison of ranking methods with varying number of captions

Ranking Method	#Captions	Accuracy (%)	Inference Time	Inference Difference	Relative Inf. Diff. (%)
Base PNP	100	62.13	0.382	0.05	0.00
	50	59.17	0.311	0.09	18.59
	5	53.52	0.225	0.09	41.10
	0	33.41	0.217	0.09	43.19
Cosine	10	54.47	0.241	0.09	36.91
	5	53.91	0.228	0.09	40.31
CLIP	10	58.11	0.274	0.08	28.27
	5	55.68	0.263	0.08	31.15
BLEU	5	54.12	0.245	0.09	35.86
BLEU+CLIP	5	60.36	0.297	0.07	22.25

Experimental results show that our proposed method has substantially lower inference time while keeping similar levels of accuracy. From 4.3 we can see the accuracy and relative inference time of different methods applied to the PNP-VQA. We can see that the accuracy of the base PNP model is the highest with 100 captions, but our proposed BLEU+CLIP method achieves similar accuracy with 5 captions only. Our method also beats the base PNP with 50 captions in terms of accuracy by using 10% of the captions only and with a higher improvement of relative accuracy.

Furthermore, all of our ranking methods resulted in substantial improvement in

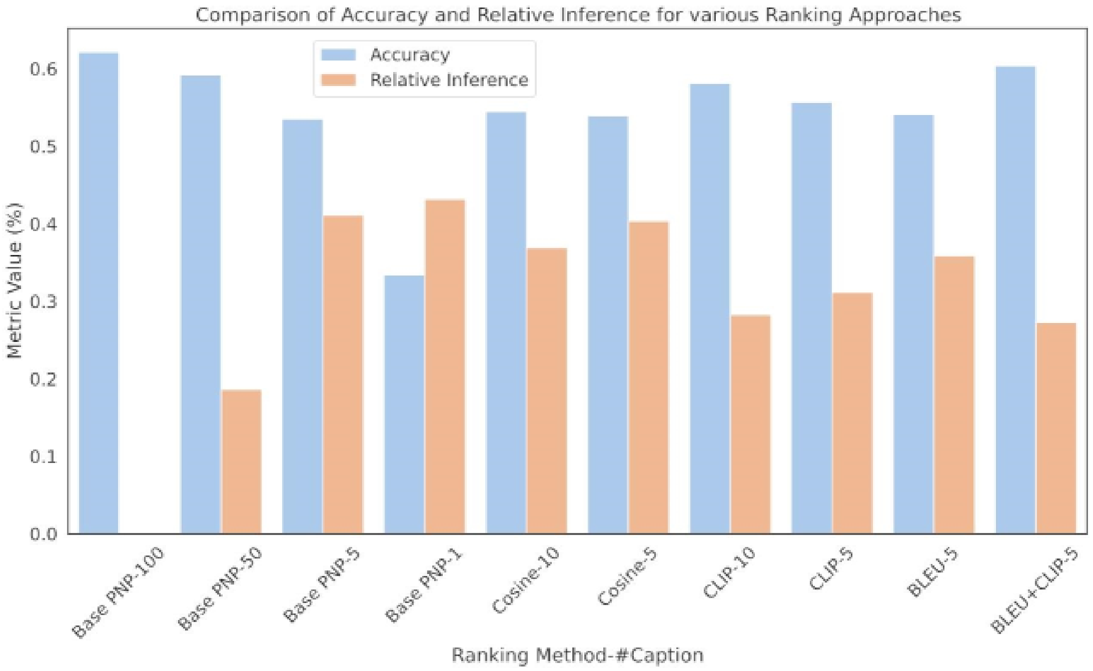


Figure 4.3: Comparison of accuracy of various ranking methods with varying number of captions along with trend lines based on ranking method

inference time. The inference time is primarily dictated by the number of captions being passed to the question-answering module which takes the question and captions as input to generate the answer as the output. Decreasing the number of captions enables the question-answering module to work on a relatively smaller set and hence, make quicker inferences.

Fig-4.4 explores the relationship between the three metrics among the various ranking methods that we proposed. Firstly, we prefer higher accuracy and high speedup which is represented by the metric relative inference difference. But, the lower the inference time, the better the method is. The base PNP using 100 captions has the highest accuracy but with the least speedup and highest inference time. Looking at the methods with higher accuracy we will see BLEU+CLIP with 5 captions has an accuracy close to base PNP followed by base PNP using 50 captions and CLIP using 10 captions. However, base-PNP with 50 captions has a very high inference time with the second lowest speedup. Our proposed method has a reasonable accuracy and speedup with a lower inference time.

4.5 Ablation Studies

The improvement in inference time was made possible through the utilization of Unigram BLEU and CLIP. By employing these techniques, we were able to assess the

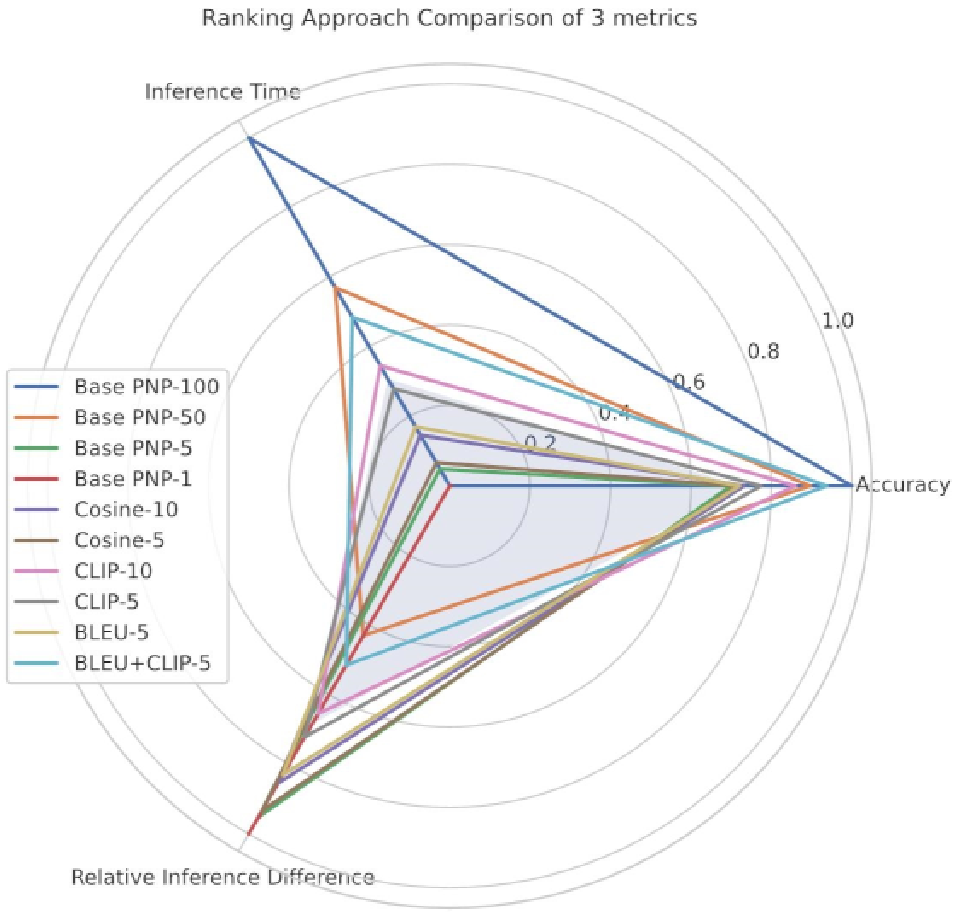


Figure 4.4: Comparison of Accuracy, Inference Time, and Relative Inference Difference after standardization for various ranking methods with varying numbers of captions

similarity between the image's caption and the given question. Consequently, we could identify the most relevant captions for generating accurate answers. Fig-4.5 illustrates a crucial point where the highest accuracy is achieved by adjusting the weight of the BLEU score. This optimization is necessary to avoid captions that merely describe the question without providing the actual answer. Based on our observations, it appears that a Unigram weight β of 60% is the optimal value to work with.

4.6 Discussion

Our work is strictly restricted by the PNP [12] architecture. In the future, we wish to explore other zero-shot architectures that have modularity. The current trend of Zero-Shot VQA along with the rise of LLMs is constantly necessitating the improvement

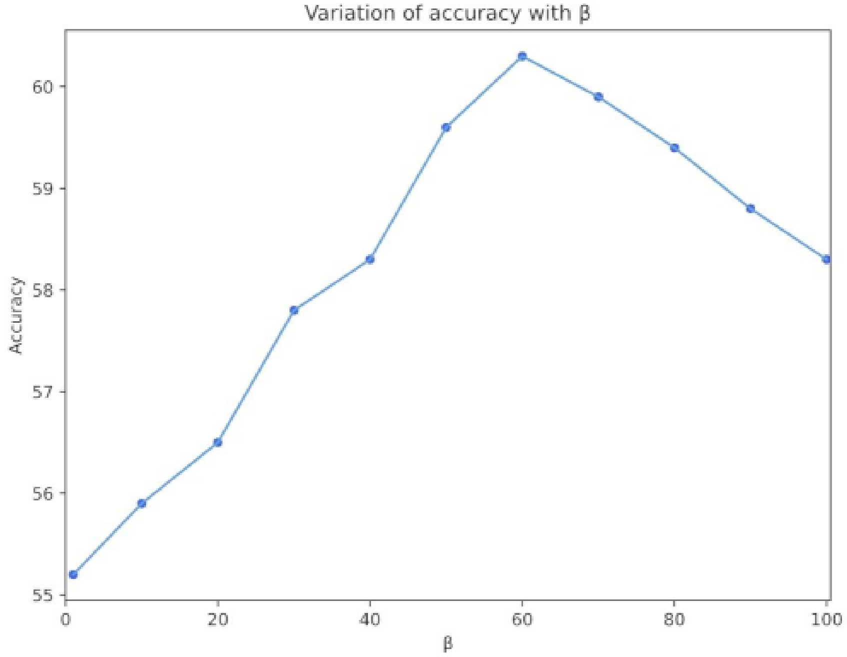


Figure 4.5: Graph of Unigram BLEU score Weight vs Accuracy

of modular architectures like PNP. However, we strongly believe that the inclusion of ranking modules will definitely help in reducing the inference outputs that are passed to the subsequent module. Inference outputs can take the form of captions, image patches, image features, etc.

Earlier ZS-VQA methods [37] heavily relied on feature extraction. While the effect of feature extraction in modern architectures has yet to be studied, feature ranking might help a certain subset of ZS-VQA models. Ranking work can be expanded to Few Shot VQA (FS-VQA) models as well. We also believe that our work failed to beat the base PNP accuracy and we wish to propose a ranking module that can not only gain a substantial speedup but also gain higher accuracy.

The Grad-CAM module [19] uses gradients to calculate which part of the image is more impactful and is analogous to attention. However, we wish to experiment with various forms of attention instead of Grad-CAM or using attention along with GRAD-CAM for a higher-quality set of image patches. Experimentations on bottom-up attention [2] replacing the Grad-CAM module have been conducted but the results were significantly worse than Grad-CAM.

Chapter 5

Conclusion and Future Work

5.1 Future Directions of Visual Robustness Analysis for VQA

There are lots of potential areas to work on to create a better framework in the future. We want to extend our idea to the addition of textual noise as well. Textual noise can be described as errors in the questions fed to the model and examples can be paraphrasing, semantic error, syntax error, etc. To imitate realistic textual error we can use typing mistakes as a type of error by associating a probability distribution to each letter being replaced by another letter. The probability associated with being replaced by a particular letter will depend on the proximity of the other letter to the pivot letter based on the keyboard layout.

Apart from textual noise, we want to explore the concept of consistency and wish to propose consistency metrics in the future. The consistency metric can be described as an evaluation metric that will quantify how consistently a model can predict the same answer with changes to the input. For instance - if we are performing binary classification and the model predicts 0,1,0,1,0 for five severity levels then it would be deemed as an inconsistent model regardless of the ground truth. Consistency has similarities and dissimilarities with robustness and we wish to explore this in the future.

Some form of preprocessing can be done before passing the input image or question to the model. The data preprocessing can be done on texts or images. Again, for a particular modality, the user can opt to not use any form of preprocessing, use white-box preprocessing i.e. preprocess the input using a technique given that the user is aware of the distribution of the noise used, or use black-box preprocessing i.e. preprocess the input without any knowledge of the distribution of noise.

Finally, we wish to train models on noisy data and make a comparative analysis of their performance. Using explainable AI, we wish to understand which parts of the images are being more focused during inferences. A major area to work on in training models with noisy data is training models on grayscale images keeping the original labels and checking the performance. We wish to explore how models who view shades of grey can associate a shade of grey with a particular color. Training colorblind models for question-answering colorblind people can also be a field of research interest.

5.2 Future Directions of Zero-Shot VQA

Zero-Shot VQA has a lot of potential areas to improve but we primarily wish to improve the multi-block modular structure instead of a single black-box LLM-based structure. Modular architectures are extensible and LLM-based units that perform well single-handedly can also be part of another modular architecture to provide a boost to accuracy. We wish to extend ZS-VQA to other domains, primarily change detection and damage assessment.

Question answering on change outputs and semantic outputs have not yet been explored and we wish to construct datasets that can produce inferential benchmarks to ZS-VQA models on such problem statements. ZS-VQA as backbone networks with VQA fine-tuning can also be used as an approach to the original VQA problem as LLM-based ZS-VQA backbones can help in answering questions that VQA networks have never seen. However, shifting to multimodal architectures resulted in multi-modal large networks to be trained in VQA by combining the modalities [3, 87]. Hopefully, we will experience rapid improvements to ZS-VQA in the future.

5.3 Conclusion

Our work introduces the proposition of using caption ranking for high-quality image-guided caption generation in the domain of zero-shot visual question answering. The proposed pipeline exploits the idea that an image is rich in data and high-quality captions can be used to utilize that data. The proposed captions are then sent to a pre-trained large language model that uses them as context to answer the question. Additionally, we propose a novel modular framework for examining the visual robustness of existing VQA methods. We added multiple realistic perturbations with adjustable levels. We aim to set our framework as a benchmark criterion for all future VQA systems to be evaluated in. As our experiments reveal the level of robustness in modern VQA methods is lacking. In a real-life scenario, it is unsurprising that contemporary methods may fail to hold up their performance. We also conclude on the fact that model size does not positively impact the robustness of a model. Additionally, we propose a novel approach to caption ranking to optimize the inference time of zero-shot VQA models. By selecting the best few captions through ranking, we can effectively reduce the computational burden without sacrificing the model's accuracy too much.

This innovative method holds promise for improving the efficiency and practicality of zero-shot VQA models in real-world applications.

Our framework can be expanded upon by adding more realistic corruption functions that a model might encounter. It can be beneficial to not just focus on visual perturbations but also on textual corruptions. A few realistic textual perturbation functions can greatly increase the scope of our framework. As for improvements to zero-shot VQA methods, future works could propose more efficient ways of querying a large language model or generating higher-quality captions that are highly related to the question.

REFERENCES

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *International Conference on Computer Vision (ICCV)*, 2015.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[3] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," *arXiv preprint arXiv:2201.12086*, 2022.

[4] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5583–5594.

[5] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "Git: A generative image-to-text transformer for vision and language," *arXiv preprint arXiv:2205.14100*, 2022.

[6] J.-H. Huang, C. D. Dao, M. Alfadly, and B. Ghanem, "A novel framework for robustness analysis of visual qa models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8449–8456.

[7] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv preprint arXiv:1903.12261*, 2019.

[8] C. E. Jimenez, O. Russakovsky, and K. Narasimhan, "Carets: A consistency and robustness evaluative test suite for vqa," *arXiv preprint arXiv:2203.07613*, 2022.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[10] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer, "How much can clip benefit vision-and-language tasks?" *arXiv preprint arXiv:2107.06383*, 2021.

[11] H. Song, L. Dong, W.-N. Zhang, T. Liu, and F. Wei, "Clip models are few-shot learners: Empirical studies on vqa and visual entailment," *arXiv preprint arXiv:2203.07190*, 2022.

[12] A. M. H. Tiong, J. Li, B. Li, S. Savarese, and S. C. Hoi, "Plug-and-play vqa: Zero-shot vqa by conjoining large pretrained models with zero training," *arXiv preprint arXiv:2210.08773*, 2022.

[13] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *arXiv preprint arXiv:2204.14198*, 2022.

[14] M. Tsimpoukelli, J. L. Menick, S. Cabi, S. Eslami, O. Vinyals, and F. Hill, "Multimodal few-shot learning with frozen language models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 200–212, 2021.

[15] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6904–6913.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] D. Khashabi, Y. Kordi, and H. Hajishirzi, "Unifiedqa-v2: Stronger generalization via broader cross-format training," *arXiv preprint arXiv:2202.12359*, 2022.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[21] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[22] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh, "Yin and yang: Balancing and answering binary visual questions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5014–5022.

[23] E. Choi, H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer, "Quac: Question answering in context," *arXiv preprint arXiv:1808.07036*, 2018.

[24] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.

[25] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," *arXiv preprint arXiv:1908.08530*, 2019.

[26] W. Wang, H. Bao, L. Dong, and F. Wei, "Vlmo: Unified vision-language pre-training with mixture-of-modality-experts," *arXiv preprint arXiv:2111.02358*, 2021.

[27] S. Barra, C. Bisogni, M. De Marsico, and S. Ricciardi, "Visual question answering: Which investigated applications?" *Pattern Recognition Letters*, vol. 151, pp. 325–331, 2021.

[28] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3608–3617.

[29] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards vqa models that can read," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra, "Visual dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[31] Z. Chen, J. Chen, Y. Geng, J. Z. Pan, Z. Yuan, and H. Chen, "Zero-shot visual question answering using knowledge graph," in *International Semantic Web Conference*. Springer, 2021, pp. 146–162.

[32] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[33] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," *arXiv preprint arXiv:1707.07328*, 2017.

[34] T. Gokhale, P. Banerjee, C. Baral, and Y. Yang, "Mutant: A training paradigm for out-of-distribution generalization in visual question answering," *arXiv preprint arXiv:2009.08566*, 2020.

[35] D. Teney and A. v. d. Hengel, "Zero-shot visual question answering," *arXiv preprint arXiv:1611.05546*, 2016.

[36] N. Karessli, Z. Akata, B. Schiele, and A. Bulling, "Gaze embeddings for zero-shot image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4525–4534.

[37] X. Wang, C. Chen, Y. Cheng, and Z. J. Wang, "Zero-shot image classification based on deep feature extraction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 2, pp. 432–444, 2016.

[38] Z. Zeng, H. Zhang, R. Lu, D. Wang, B. Chen, and Z. Wang, "Conzic: Controllable zero-shot image captioning by sampling-based polishing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 465–23 476.

[39] Y. Fu, T. Xiang, Y.-G. Jiang, X. Xue, L. Sigal, and S. Gong, "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 112–125, 2018.

[40] R. Cadene, C. Dancette, M. Cord, D. Parikh *et al.*, "Rubi: Reducing unimodal biases for visual question answering," *Advances in neural information processing systems*, vol. 32, 2019.

[41] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[42] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2612–2620.

[43] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; look and answer: Overcoming priors for visual question answering," in *Proceedings*

of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4971–4980.

[44] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 21–29.

[45] A. Agrawal, A. Kembhavi, D. Batra, and D. Parikh, "C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset," *arXiv preprint arXiv:1704.08243*, 2017.

[46] S. Ramakrishnan, A. Agrawal, and S. Lee, "Overcoming language priors in visual question answering with adversarial regularization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[47] C. Clark, M. Yatskar, and L. Zettlemoyer, "Don't take the easy way out: Ensemble based methods for avoiding known dataset biases," *arXiv preprint arXiv:1909.03683*, 2019.

[48] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *Advances in neural information processing systems*, vol. 29, 2016.

[49] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6281–6290.

[50] A. K. Singh, A. Mishra, S. Shekhar, and A. Chakraborty, "From strings to things: Knowledge-enabled vqa model that can read and reason," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4602–4612.

[51] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," *arXiv preprint arXiv:1908.07490*, 2019.

[52] W. Li, C. Gao, G. Niu, X. Xiao, H. Liu, J. Liu, H. Wu, and H. Wang, "Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning," *arXiv preprint arXiv:2012.15409*, 2020.

[53] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 4904–4916. [Online]. Available: https://proceedings.mlr.press/v139/jia21b.html

[54] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," *Advances in neural information processing systems*, vol. 34, pp. 9694–9705, 2021.

[55] W. Jin, Y. Cheng, Y. Shen, W. Chen, and X. Ren, "A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2763–2775. [Online]. Available: https://aclanthology.org/2022.acl-long.197

[56] W. Dai, L. Hou, L. Shang, X. Jiang, Q. Liu, and P. Fung, "Enabling multimodal generation on CLIP via vision-language knowledge distillation," in *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2383–2395. [Online]. Available: https://aclanthology.org/2022.findings-acl.187

[57] P. Banerjee, T. Gokhale, Y. Yang, and C. Baral, "WeaQA: Weak supervision via captions for visual question answering," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 3420–3435. [Online]. Available: https://aclanthology.org/2021.findings-acl.302

[58] S. Changpinyo, D. Kukliansky, I. Szpektor, X. Chen, N. Ding, and R. Soricut, "All you may need for vqa are image captions," *arXiv preprint arXiv:2205.01883*, 2022.

[59] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

[65] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[66] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[67] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[68] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[69] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6700–6709.

[70] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[71] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach, "GPT-NeoX-20B: An open-source autoregressive language model," in *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*. virtual+Dublin: Association for Computational Linguistics, May 2022, pp. 95–136. [Online]. Available: https://aclanthology.org/2022.bigscience-1.9

[72] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model," https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[73] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[74] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, "Pythia v0. 1: the winning entry to the vqa challenge 2018," *arXiv preprint arXiv:1807.09956*, 2018.

[75] Z. Huang, Z. Zeng, B. Liu, D. Fu, and J. Fu, "Pixel-bert: Aligning image pixels with text by deep multi-modal transformers," *arXiv preprint arXiv:2004.00849*, 2020.

[76] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer." *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[77] Z. Yang, Z. Gan, J. Wang, X. Hu, Y. Lu, Z. Liu, and L. Wang, "An empirical study of gpt-3 for few-shot knowledge-based vqa," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 3081–3089.

[78] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[79] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.

[80] A. Awad, "Denoising images corrupted with impulse, gaussian, or a mixture of impulse and gaussian noise," *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 746–753, 2019.

[81] S. W. Hasinoff, "Photon, poisson noise." *Computer Vision, A Reference Guide*, vol. 4, p. 16, 2014.

[82] A. Maity, A. Pattanaik, S. Sagnika, and S. Pani, "A comparative study on approaches to speckle noise reduction in images," in *2015 International Conference on Computational Intelligence and Networks*. IEEE, 2015, pp. 148–155.

[83] A. Maity and R. Chatterjee, "Impulsive noise in images: a brief review," *Computer Vision Graphics and Image Processing*, vol. 4, pp. 6–15, 2018.

[84] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.

[85] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," *arXiv preprint arXiv:2202.03052*, 2022.

[86] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[87] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

APPENDICES

A Predictions by BLIP [3] on Color-related Questions

All the images in this appendix, are from VQAv2 [15] on color-related questions. In the given cases, the model correctly predicts the answer at severity level-4 (first image) but mispredicts at severity level-5 (second image). However, the mispredictions occur primarily due to inconsistent answer generation and mislabeling problems.

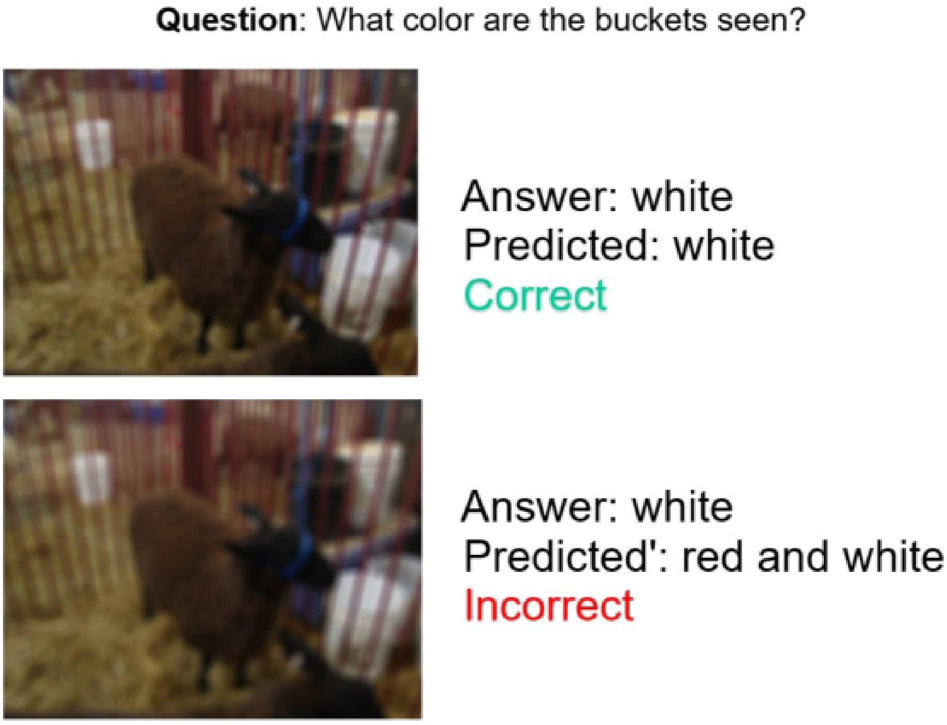

Question: What color are the buckets seen?
Answer: white
Predicted: white
Correct
Answer: white
Predicted': red and white
Incorrect

Figure A.1: Model mispredicts due to mixture of the red bars with the white color of the bucket. However, the actual color of the bucket in severity level-5 is recognizable by a human.

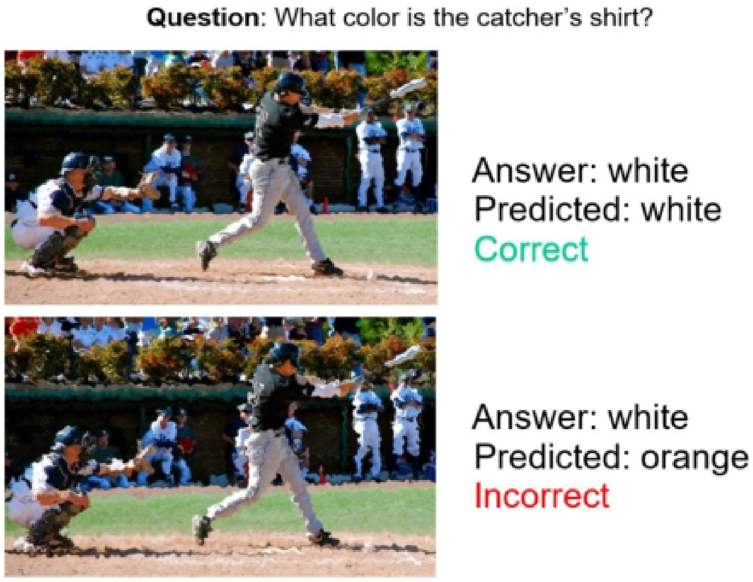


Figure A.2: There is no logical reasoning behind this misprediction and it can be simply deemed as model inconsistency.

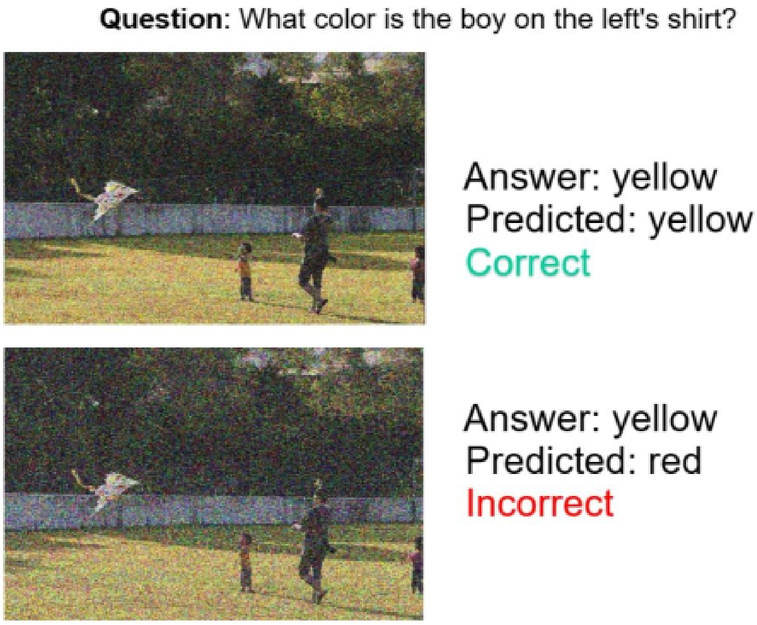


Figure A.3: Model couldn’t predict correctly as the added noise makes it seem like the color has been changed. Humans are divided on the answer to this question. However, this isn’t a mislabeling issue.

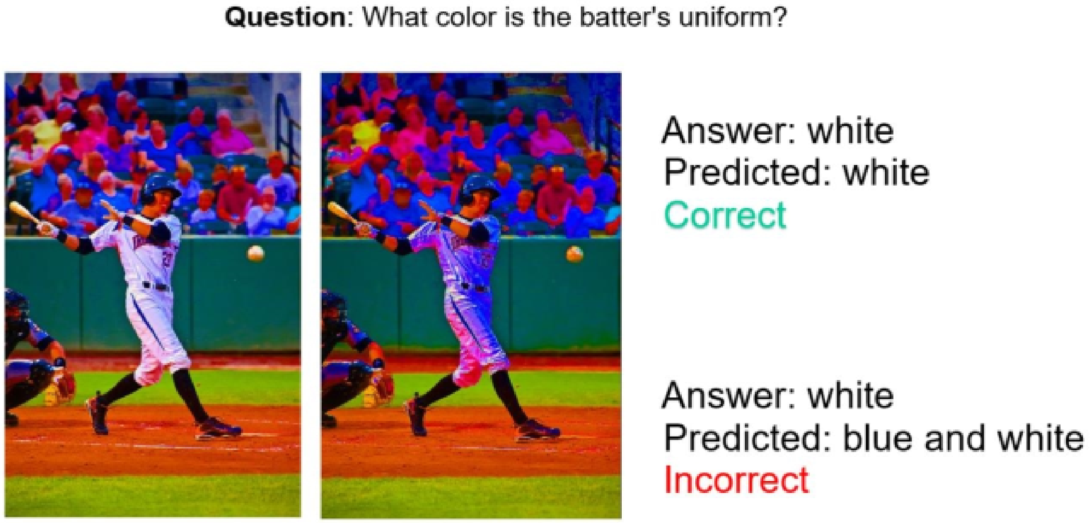

Question: What color is the batter's uniform?
Answer: white
Predicted: white
Correct
Answer: white
Predicted: blue and white
Incorrect

Figure A.4: The prediction of the model is correct in this scenario and the ground truth should have been different for severity level-5. Hence, this is a mislabeling issue.

B Predictions by BLIP [3] on Counting Questions

Similar to Appendix-A, but for counting questions.

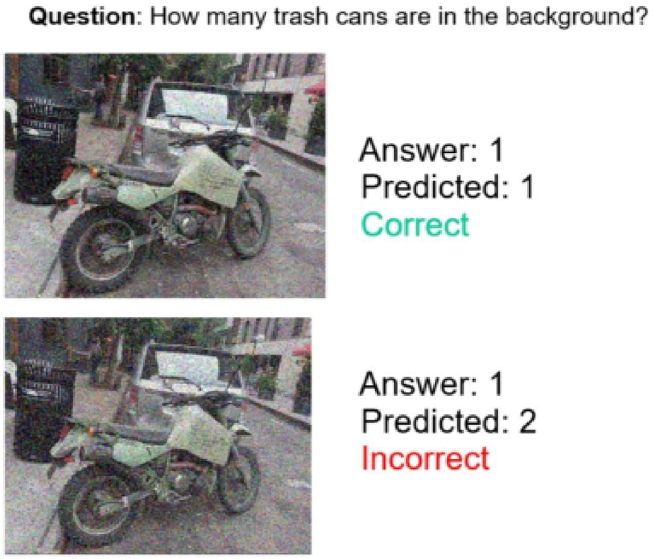


Figure B.1: Model misprediction due to inconsistency. The question is easily answerable by a human.

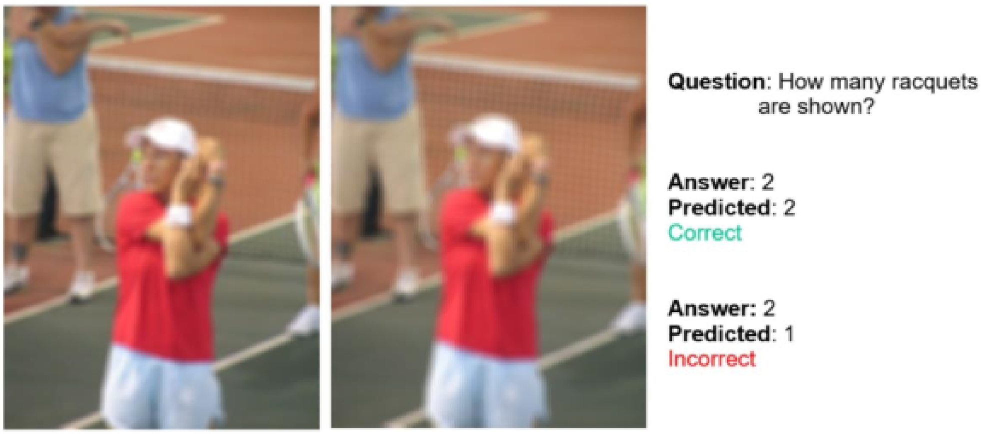


Figure B.2: Answering the question based on the severity level-5 image is difficult even for a human.

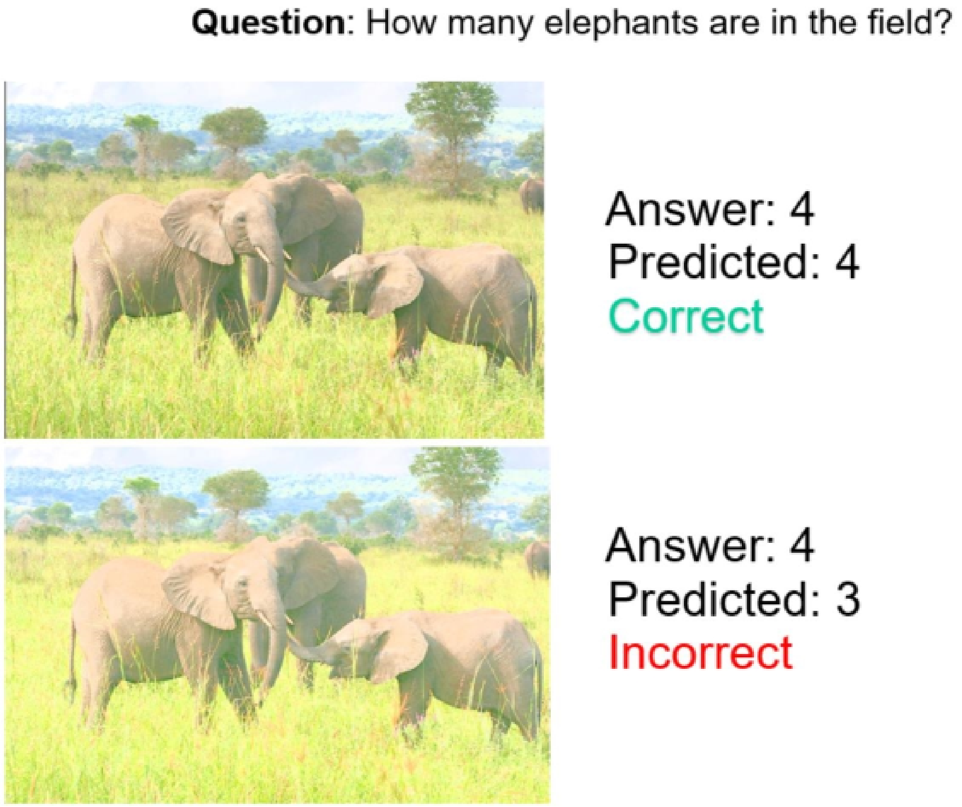


Figure B.3: Due to added brightness the model merges two elephant bodies into a single one.

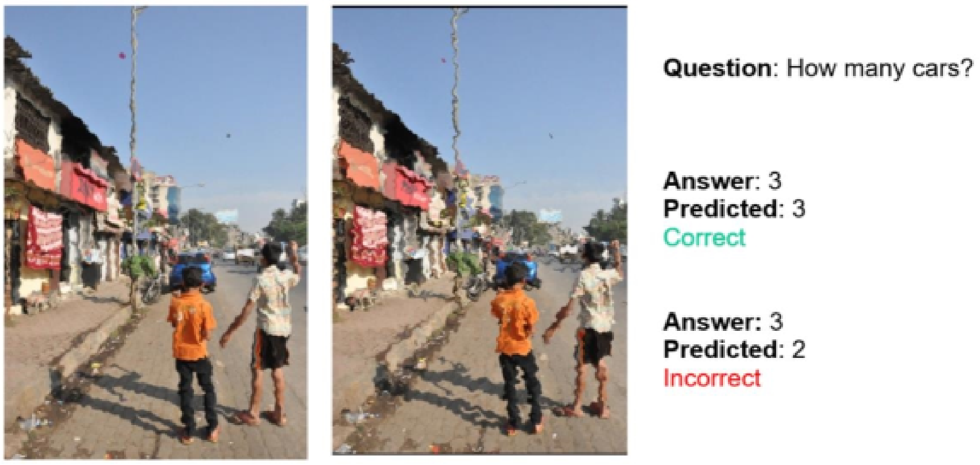


Figure B.4: Mislabeling problem; a human will also have the same answer as the model for severity level-5.