

UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

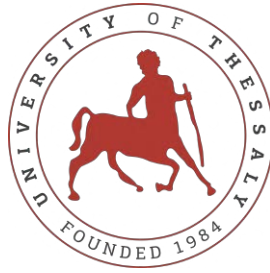
Cross-Modal Learning for Image Captioning

Diploma Thesis

Theodora Poultsidou

Supervisor: Gerasimos Potamianos

September 2025



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Cross-Modal Learning for Image Captioning

Diploma Thesis

Theodora Poultsidou

Supervisor: Gerasimos Potamianos

September 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Διατροφική Εκμάθηση για Επεξήγηση Εικόνων

Διπλωματική Εργασία

Θεοδώρα Πουλτσίδου

Επιβλέπων: Γεράσιμος Ποταμιάνος

Σεπτέμβριος 2025

Approved by the Examination Committee:

Supervisor **Gerasimos Potamianos**

Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Daskalopulu Aspassia**

Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Dimitrios Rafailidis**

Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

I would like to express my deep appreciation to my supervisor, Assoc. Prof. Gerasimos Potamianos, for his continuous guidance and the detailed feedback that greatly improved the quality of this thesis. I am also thankful to the committee members, Assoc. Prof. Aspasia Daskalopoulou and Assoc. Prof. Dimitrios Rafailidis, for dedicating their time to attend the presentation and review my work.

Finally, I would like to thank my family and friends for always being there, whether it was by encouraging me to keep going, distracting me at the right moments, or simply reminding me that life exists beyond this thesis.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Theodora Poultsidou

Diploma Thesis

Cross-Modal Learning for Image Captioning

Theodora Poultsidou

Abstract

This thesis examines image captioning through a comparison of classical Encoder-Decoder models and modern pretrained Vision-Language models (VLMs). The first part looks at models trained from scratch, testing variations in encoders, attention mechanisms, word embeddings, recurrent decoders, and decoding methods. Among these, the strongest setup combined a transformer-based encoder with attention, trainable embeddings, a gated recurrent decoder, and beam search. The second part evaluates large-scale pretrained VLMs in a zero-shot setting. These models show strong results without dataset-specific training and often outperform the systems trained from scratch, though they can still produce small factual errors. This naturally extends to prompt-based captioning. Here, the same image is described in different ways depending on the instruction, for example, a neutral caption, a creative one, or an accessibility-focused one. While prompts that match the dataset's style score best under traditional metrics, alternative prompts often give richer and more useful descriptions. This highlights both the flexibility of promptable models and the limitations of current evaluation methods.

Overall, the findings suggest that while traditional captioning pipelines remain useful for controlled experiments, general-purpose VLMs are quickly surpassing them. This reflects a broader trend in AI, where large pretrained models increasingly replace task-specific architectures, and image captioning is no exception.

Keywords:

Image captioning, cross-modal learning, encoder-decoder, attention mechanisms, vision-language models, prompt-based captioning

Διπλωματική Εργασία

Διατροφική Εκμάθηση για Επεξήγηση Εικόνων

Θεοδώρα Πουλτσίδου

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει το θέμα της επεξήγησης εικόνων μέσα από τη σύγκριση κλασικών μοντέλων τύπου Encoder–Decoder και σύγχρονων προεκπαιδευμένων μοντέλων όρασης-γλώσσας (Vision-Language Models). Στο πρώτο μέρος μελετώνται μοντέλα που εκπαιδεύτηκαν από την αρχή, δοκιμάζοντας διαφορετικές επιλογές σε κωδικοποιητές εικόνας, μηχανισμούς προσοχής, ενσωματώσεις λέξεων, επαναληπτικούς αποκωδικοποιητές και μεθόδους αποκωδικοποίησης. Η καλύτερη διαμόρφωση συνδύασε κωδικοποιητή τύπου transformer με μηχανισμό προσοχής, εκπαιδύσιμες ενσωματώσεις, αποκωδικοποιητή GRU και beam search. Στο δεύτερο μέρος αξιολογούνται μεγάλα προεκπαιδευμένα VLMs σε συνθήκες zero-shot, δηλαδή χωρίς επιπλέον εκπαίδευση στο συγκεκριμένο σύνολο δεδομένων. Τα μοντέλα αυτά επιτυγχάνουν υψηλές επιδόσεις και συχνά ξεπερνούν τα συστήματα που εκπαιδεύτηκαν από την αρχή, αν και μπορεί να εμφανίσουν μικρά λάθη. Η ανάλυση επεκτείνεται στη δημιουργία περιγραφών με προτροπές (prompt-based captioning), όπου η ίδια εικόνα μπορεί να περιγραφεί με διαφορετικούς τρόπους, ανάλογα με την οδηγία, για παράδειγμα, με μια ουδέτερη, μια δημιουργική ή μια προσανατολισμένη στην προσβασιμότητα περιγραφή. Αν και οι προτροπές που ταιριάζουν στο ύφος του συνόλου δεδομένων αποδίδουν καλύτερα στις παραδοσιακές μετρικές, οι εναλλακτικές οδηγούν συχνά σε πιο πλούσιες και χρηστικές περιγραφές. Αυτό αναδεικνύει τόσο την ευελιξία των promptable μοντέλων όσο και τους περιορισμούς των τρεχουσών μεθόδων αξιολόγησης.

Συνολικά, τα αποτελέσματα δείχνουν ότι, ενώ οι κλασικές προσεγγίσεις παραμένουν χρήσιμες για ελεγχόμενα πειράματα, τα γενικού σκοπού προεκπαιδευμένα VLMs τις ξεπερνούν ταχύτατα. Η τάση αυτή αντανακλά μια γενικότερη πορεία στην Τεχνητή Νοημοσύνη, όπου τα μεγάλα προεκπαιδευμένα μοντέλα αντικαθιστούν ολοένα και περισσότερο τις εξειδικευμένες αρχιτεκτονικές, και η επεξήγηση εικόνων δεν αποτελεί εξαίρεση.

Λέξεις-κλειδιά:

Επεξήγηση εικόνων, διατροφική εκμάθηση, encoder-decoder, μηχανισμοί προσοχής, μοντέλα όρασης-γλώσσας, περιγραφές με προτροπές

Table of contents

Acknowledgements	ix
Abstract	xii
Περίληψη	xiii
Table of contents	xv
List of figures	xix
List of tables	xxi
Abbreviations	xxiii
1 Introduction	1
1.1 Thesis Scope and Objectives	1
1.2 Contributions	2
1.3 Thesis Structure	3
2 Related Work	5
2.1 Before Deep-Learning: Templates and Retrieval	5
2.2 Neural Encoder-Decoder Era	6
2.3 Attention Mechanisms and Stronger Decoders	7
2.4 Training Strategies and Decoding	7
2.5 From Captioners to General Vision-Language Models	8
2.6 Datasets and Evaluation	10
2.7 Positioning of this Thesis	10

3	Dataset	13
3.1	The Flickr8k Dataset	13
3.2	Caption Characteristics	14
3.2.1	Length Distribution	14
3.2.2	Vocabulary Characteristics	15
3.2.3	Frequent Words	15
3.2.4	Flickr8k Dataset Example	16
3.3	Text Preprocessing	16
3.3.1	Preprocessing Flow	17
4	Building Captions with Encoder-Decoder Models	19
4.1	Architecture Overview	19
4.2	Image Encoders	21
4.2.1	ResNet-50	21
4.2.2	EfficientNetB0	22
4.2.3	Vision Transformer	23
4.3	Word Embeddings	24
4.4	Training Process	25
4.5	Attention Mechanisms and Fusion	26
4.5.1	Overall Architecture	27
4.5.2	Computation Process	27
4.5.3	Attention Variants	28
4.5.4	No-Attention (Global Fusion)	29
4.5.5	Visualizations	29
4.6	Inference and Decoding	30
4.6.1	Decoder Architecture	31
4.6.2	Greedy Decoding	31
4.6.3	Beam Search Decoding	32
5	Better Captions with Vision-Language Models	35
5.1	Models Overview	36
5.1.1	Bootstrapped Language-Image Pretraining (BLIP)	36
5.1.2	Generative Image-to-Text Transformer (GIT)	37

5.1.3	BLIP-2	37
5.1.4	Large Language and Vision Assistant (LLaVA)	38
6	Evaluation Metrics	39
6.1	BLEU: Bilingual Evaluation Understudy	40
6.2	METEOR: Metric for Evaluation of Translation with Explicit ORdering . .	41
6.3	CIDER: Consensus-Based Evaluation	42
6.4	ROUGE-L: Longest Common Subsequence	43
6.5	Embedding-Based Metrics: BERTScore and CLIPScore	43
6.5.1	BERTScore	43
6.5.2	CLIPScore	44
6.6	Why all these Metrics	44
7	Experiments and Results	47
7.1	Evaluation of Classical Encoder-Decoder Architectures	47
7.1.1	Comparison of Image Encoders	47
7.1.2	Attention and Fusion Strategies	49
7.1.3	Training Dynamics	52
7.1.4	Word Embeddings: Trainable vs. Pretrained	53
7.1.5	Decoder and Search Strategies	54
7.2	Evaluation of Pretrained Vision-Language Models	57
7.2.1	Prompt-Based Captioning with LLaVA	58
8	Conclusions and Future Work	65
8.1	Conclusions	65
8.2	Future Work	66
	Bibliography	69
	APPENDICES	79
A	Additional Material	81
A.1	Additional Attention Visualizations	81
A.2	Additional Prompt-Based Examples	84
A.3	LLM as a Judge Evaluation Criteria Prompt	86

List of figures

3.1	Distribution of caption lengths in the Flickr8k dataset.	14
3.2	Top 20 most frequent words in the Flickr8k dataset (excluding function-words).	15
3.3	Example image from Flickr8k.	16
4.1	General encoder-decoder training pipeline for image captioning, consisting of (1) an encoder for image feature extraction, (2) caption preprocessing to embed the input text, and (3) a decoder that generates the output caption.	20
4.2	ResNet-50 encoder: It outputs a global feature vector or a spatial feature map for attention (figure adapted from [1]).	22
4.3	EfficientNetB0 encoder: It outputs a global vector or spatial map for attention (figure adapted from [2]).	22
4.4	ViT encoder: images are split into patches, embedded, positionally encoded, and processed through Transformer layers (figure adapted from [3]).	23
4.5	Decoder training with teacher forcing: the model conditions on the previous ground-truth word (shown in purple) rather than its own prediction (shown in green).	26
4.6	Attention in the training pipeline of image captioning. At each step, the decoder can focus on different regions of the feature map.	27
4.7	Step-by-step computation of attention at decoding step t	28
4.8	Attention maps for selected words (Bahdanau Attention).	30
4.9	Illustration of the inference process.	31
4.10	Greedy decoding always selects the highest-probability word.	32
4.11	Beam search explores multiple candidates and chooses the best overall sequence.	33
7.1	Image used to compare captions generated by different visual encoders.	48

7.2	Captions generated by different fusion strategies.	49
7.3	Word-by-word visualization of additive attention using ViT features.	50
7.4	Word-by-word visualization of Luong attention using ViT features.	51
7.5	Training dynamics of the ViT + Additive Attention + GloVe model. Validation loss stabilizes around epoch 3 or 4 while accuracy plateaus shortly afterwards, indicating convergence without overfitting.	52
7.6	Training dynamics of the ViT + Luong Attention + GloVe model. Validation loss stabilizes again around epoch 3 or 4 while accuracy plateaus shortly afterwards, indicating convergence (slower than additive attention) without overfitting.	52
7.7	Image used to compare the effect of embedding type on generated captions.	54
7.8	Caption outputs for different beam widths on the same image.	55
7.9	Image used to compare captioning styles of BLIP, BLIP-2, and GIT.	57
7.10	Image used for prompt-based generation with LLaVA.	60
A.1	Example 1: Predicted caption “ <i>a young girl does a handstand on the beach.</i> ”	82
A.2	Example 2: Predicted caption “ <i>a black dog is running on the beach next to the ocean.</i> ”	82
A.3	Example 3 (failure case): Predicted caption “ <i>a brown dog is jumping in the air to catch a brown dog.</i> ”	83
A.4	Image used for prompt-based generation with LLaVA.	84
A.5	Image used for prompt-based generation with LLaVA.	85

List of tables

3.1	Overview of the Flickr8k dataset	13
3.2	Vocabulary statistics before preprocessing	15
4.1	Embedding configurations tested	24
4.2	Fusion methods tested. Both attention and concatenation are late fusion approaches	29
5.1	Overview of the vision-language models used in this study	36
7.1	Performance of image encoders with identical downstream setup	48
7.2	Performance of attention and fusion strategies (ViT encoder)	49
7.3	Performance of embedding types (ViT + Additive Attention + GRU + Greedy Decoding)	53
7.4	Effect of decoding strategy (ViT + GRU + Additive Attention)	55
7.5	LSTM vs. GRU decoder comparison (ViT + Additive Attention)	55
7.6	Decoding time per caption for different beam widths (example of Figure 7.8)	56
7.7	Zero-shot performance of pretrained VLMs	57
7.8	Prompted captioning results using LLaVA on 1000 Flickr8k images	61
7.9	Average LLM-as-a-judge evaluation across different prompt types (scores 1–5)	62
7.10	Accessibility Fit and Social Media Fit scores across different prompt types (scores 1–5)	63

Abbreviations

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BLIP	Bootstrapped Language-Image Pretraining
CIDEr	Consensus-based Image Description Evaluation
CLIP	Contrastive Language-Image Pretraining
CNN	Convolutional Neural Network
GIT	Generative Image-to-Text Transformer
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
LLaVA	Large Language and Vision Assistant
LSTM	Long Short-Term Memory
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
TF-IDF	Term Frequency-Inverse Document Frequency
ViT	Vision Transformer
VLM	Vision-Language Model
VQA	Visual Question Answering

Chapter 1

Introduction

When we look at an image, describing it in words is natural and effortless. We can instantly identify the objects, understand their relationships, and formulate them into a meaningful sentence. For a machine, however, this is a complex task that requires the integration of two challenging domains: *computer vision*, to extract and interpret visual information, and *natural language processing*, to generate coherent and semantically accurate descriptions. This task is known as **image captioning**.

Image captioning has significant real-world applications. It can make visual content more accessible to people with visual disabilities, improve the organisation and search of large image collections, and support human-AI interaction in fields such as robotics, autonomous vehicles, and assistive technologies. Because it demands an understanding of both visual and text modalities, it is also a standard problem for studying *cross-modal learning*.

1.1 Thesis Scope and Objectives

This thesis focuses on the task of image captioning and aims to study, implement, and compare both classical and modern approaches. The work is split into two main parts: first, we construct and evaluate a series of classical encoder-decoder models, experimenting with different image encoders (ResNet50, EfficientNetB0, ViT), attention mechanisms (additive [4], Luong [5]), fusion methods, and decoders (GRU, LSTM) trained on the Flickr8k dataset [6]. Then, we explore the capabilities of large pretrained vision-language models (BLIP [7], GIT [8], BLIP-2 [9], LLaVA [10]), including their performance in zero-shot captioning and their ability to follow prompts to generate stylistically or semantically diverse captions.

The central research questions of this thesis include:

- How do different design choices (encoder, decoder, attention, decoding strategy) affect the quality of captions in traditional models?
- Can pretrained vision-language models outperform custom models without any fine-tuning?
- Do promptable models offer more flexible and purpose-specific captions, and how can we evaluate them fairly?

1.2 Contributions

To answer these questions, the main contributions of this thesis are the following. We have:

1. Built and trained multiple encoder-decoder models on Flickr8k, varying the encoder (ResNet50, EfficientNetB0, ViT), attention type (Bahdanau, Luong), fusion method, embeddings (trainable vs. GloVe [11]), and decoder (LSTM vs. GRU), and tested greedy vs. beam search decoding.
2. Evaluated three large-scale VLMs (BLIP, GIT, and BLIP-2) in zero-shot mode on Flickr8k.
3. Designed and ran prompt-based experiments with LLaVA using eight different instructions, targeting styles from basic description to accessibility-focused and creative captions.
4. Analyzed outputs with both traditional metrics (BLEU, METEOR, ROUGE-L, CIDEr) and semantic/vision-language metrics (BERTScore [12], CLIPScore [13]), highlighting where usual scores fail to capture quality.
5. Added an evaluation step using GPT-5 [14] as an LLM judge, scoring captions on accuracy, relevance, specificity, conciseness, fluency, and purpose-fit, to provide a more human-aligned view of quality than automatic metrics.

1.3 Thesis Structure

- Chapter 2 presents related work and background on image captioning, encoder-decoder architectures, attention mechanisms, and pretrained models.
- Chapter 3 introduces the dataset, preprocessing steps, and experimental setup.
- Chapter 4 analyzes classical encoder-decoder models, testing various combinations of encoders, decoders, and attention mechanisms.
- Chapter 5 presents large vision-language models, their capabilities, and the idea of promptable captioning.
- Chapter 6 discusses the evaluation metrics used throughout this work, including both traditional and modern semantic metrics.
- Chapter 7 presents all experimental results and comparisons.
- Chapter 8 summarizes the conclusions and suggests directions for future work.

Chapter 2

Related Work

Image captioning has evolved rapidly over the past decade. The field has moved from simple template-based approaches, which relied on predefined sentence structures, to deep-learning methods that jointly model visual and text information. Today, large vision-language models trained on massive datasets dominate the landscape, offering impressive zero-shot capabilities and the ability to follow user instructions. This chapter reviews the main stages of this development, highlighting the models and techniques most relevant to the experiments carried out in this thesis.

2.1 Before Deep-Learning: Templates and Retrieval

Early approaches to image captioning often relied on predefined sentence patterns, known as *template-based* methods. After detecting a few objects or actions in the scene, the system would simply fill in the blanks of a fixed template. A well-known example is the work of Farhadi *et al.* [15], who represented an image as a structured triple (*object, action, scene*) and then converted that representation into a caption. For instance, if the system detected (*dog, running, park*), the output could be a sentence like “*The dog is running in the park.*” This produced correct sentences, but the language was repetitive and heavily constrained by the fixed templates and the rather weak object detectors of that era.

A second approach was *retrieval-based*, where captions were borrowed from similar images instead of being generated from scratch. Instead of generating a caption from scratch, these systems reused captions that had already been written by humans. Ordoñez *et al.* [16]

collected a very large set of images with their corresponding captions, and for a new query image they simply retrieved the caption of a visually similar photo using global image features. Sometimes minor adjustments were applied to the caption to better fit the new image, but in most cases it was copied as is. This approach ensured fluent and natural sentences, but it could also generate misleading results, if the retrieved images were not semantically close to the query.

Both template-based and retrieval-based methods showed that producing natural language for images was possible, but they also showed key limitations. Template systems lacked flexibility, and retrieval systems lacked true visual understanding. These shortcomings motivated the development of neural models that could both *understand* images and *generate* new captions more flexibly.

2.2 Neural Encoder-Decoder Era

With the rise of deep-learning, image captioning began to use end-to-end neural models. One influential example was the “Show and Tell” system by Vinyals *et al.* [17], which approached captioning in the same way as machine translation. A convolutional network produced a visual feature vector, and a recurrent network such as an LSTM or GRU generated the caption step by step. At the same time, Karpathy and Fei-Fei [18] proposed a model that linked regions of an image with fragments of a sentence in a shared embedding space, and then generated captions using those alignments.

Together, these works introduced the encoder-decoder approach: a visual encoder to extract image features, a sequential decoder to produce text, and training on paired images and captions. The idea was simple but also exhibited some clear drawbacks. Early models only passed a single global context vector to the decoder, which made it difficult to describe smaller objects or fine details. In addition, training used teacher forcing, feeding the correct word at every step, while at test time the model had to rely on its own predictions. This gap, known as exposure bias, often caused captions to drift away from the ground truth.

These challenges led to two major lines of improvement: attention mechanisms, which al-

lowed the model to focus on different parts of the image as each word was generated, and alternative training and decoding methods to reduce exposure bias.

2.3 Attention Mechanisms and Stronger Decoders

The first encoder-decoder models showed promise, but their use of a single global context vector limited how well they could describe detailed scenes. To address this, Xu *et al.* [19] introduced the idea of visual attention in their “Show, Attend and Tell” model. Instead of relying on one fixed representation of the image, the decoder could shift its focus to different regions at each step of the caption. This meant that when predicting a word like “dog,” the model could attend to the part of the image containing the dog, and later move its focus to the background when generating words about the scene. Attention quickly became a standard component because it improved both accuracy and interpretability, as researchers could visualize what the model was looking at while producing each word.

Building on this idea, later work explored more structured forms of attention. Anderson *et al.* [20] proposed a bottom-up and top-down approach, where the model attended not to uniform grid regions but to object proposals detected by a Faster R-CNN network. This allowed the captions to reference objects more precisely and improved performance on benchmark datasets.

Attention also opened the door to stronger decoders. Instead of simply feeding attended features into an LSTM, researchers experimented with stacking multiple layers, using gating mechanisms to control how much visual versus text information was used, and eventually adopting Transformer-based architectures [21]. These developments shifted captioning from a simple CNN-RNN pipeline into a more flexible framework where visual grounding and language generation were closely integrated.

2.4 Training Strategies and Decoding

Training captioning models is not only about the architecture but also about how the decoder learns and how captions are generated at test time. The earliest systems relied on a simple

strategy called *teacher forcing* [22]: at each step, the decoder was given the ground-truth word as input. This made training stable, but it created a gap with inference, where the model had to rely on its own predictions instead. As mentioned earlier, this mismatch, known as exposure bias, often caused captions to drift away from the correct description.

Several solutions were proposed to close this gap. One of the most influential was self-critical sequence training (SCST) by Rennie *et al.* [23], which directly optimized sequence-level metrics such as CIDEr [24] instead of just likelihood. The idea was to let the model generate its own captions, compare them to a baseline (often greedy decoding), and update the parameters based on whether the sampled caption performed better. This reinforcement learning approach helped align training with the actual evaluation metrics used in practice.

Decoding strategies also played an important role. The simplest method, greedy decoding, selects the most likely word at each step, but this can lead to repetitive or overly safe captions. Beam search [25] was introduced as a more robust alternative: instead of committing to a single word at each step, the model keeps track of multiple candidate sequences and expands the most promising ones. While beam search generally improves caption quality, it comes with higher computational cost and sometimes produces captions that sound fluent but lack diversity.

These training and decoding refinements were crucial in making encoder-decoder models more reliable and competitive, and they set the stage for later developments that relied heavily on attention and large-scale pretraining.

2.5 From Captioners to General Vision-Language Models

While encoder-decoder models with attention made captioning much stronger, they were still trained on relatively small datasets with a narrow focus: generating short descriptions for images. The next big step came with large-scale pretraining on paired image-text data, which opened the way for more general vision-language models.

A key milestone was CLIP by Radford *et al.* [26], which trained on hundreds of millions

of image-text pairs collected from the web. Instead of producing captions directly, CLIP learned a joint embedding space where images and text could be compared. This simple idea turned out to be very powerful: a model trained in this way could be used for tasks such as zero-shot classification and retrieval, showing that large-scale image-text pretraining could generalize far beyond captioning.

Building on this trend, models like BLIP [7], BLIP-2 [9], and GIT [8] combined large-scale pretraining with caption generation. BLIP introduced a bootstrapping method to filter noisy web data and improve alignment, while BLIP-2 added a clever way to connect frozen language models (such as BERT [27] or GPT-style decoders [14, 28]) to vision encoders through lightweight adapters. GIT, on the other hand, showed that with enough data and transformer layers, captioning could be scaled up in a simple but effective way. Other key models include OSCAR [29], which used object tags to guide caption generation, and VinVL [30], which improved visual features and became a strong baseline for many benchmarks. SimVLM [31] simplified the design with a single transformer and showed that scaling data alone could give very strong results. Larger multimodal models like ALIGN [32] and Flamingo [33] trained on billions of image-text pairs and demonstrated that captioning could be treated as just one capability within a broader vision-language system.

More recently, instruction-tuned models have brought captioning closer to interactive use. Systems such as LLaVA [34], InstructBLIP [35], and Kosmos-2 [36] can generate not only plain captions but also context-dependent descriptions, follow prompts, or answer questions about an image. For example, instead of just saying “*A man riding a bicycle*”, such a model could adapt the style if asked to “*Describe the scene as if for social media*” or “*Give a detailed caption for accessibility.*”

These advances mark a shift from task-specific captioning systems to general-purpose VLMs that treat captioning as just one capability among many. They also set the stage for this thesis, which compares classical encoder-decoder models with modern VLMs and explores the use of prompting to produce more flexible captions.

2.6 Datasets and Evaluation

Progress in image captioning has always been tied to the datasets used for training and evaluation. Early work relied on small, curated collections of images, but these were too limited to train modern neural models. The introduction of Flickr8k [6], Flickr30k [37], and MSCOCO [38] provided the first widely used standards. Each image in these datasets is paired with multiple human-written captions, which is important because a single picture can often be described in many valid ways. Having several references allows models to be evaluated more fairly.

Evaluating captions, however, has remained a challenge. Standard metrics were borrowed from machine translation and summarization. BLEU [39] counts overlapping n-grams between a generated caption and the references, while ROUGE [40] does something similar with recall in mind. METEOR [41] improved on these by adding stemming and synonym matching, and CIDEr [24] became popular in captioning because it weights n-grams by how specific they are in the dataset. Each of these metrics captures part of what makes a caption good, but none can fully reflect human judgments of fluency and relevance.

Because of these limits, researchers have increasingly turned to human evaluation or, more recently, to large language models (LLMs) used as evaluators [42, 28]. An LLM can rate captions on multiple dimensions such as accuracy, detail, and fluency [43], and can also judge whether a caption fits a particular purpose (for example, accessibility text versus a social media post). This thesis also adopts that approach, combining traditional automatic scores with LLM-based judgments to obtain a more complete picture of caption quality.

2.7 Positioning of this Thesis

The aforementioned works show how image captioning has developed step by step: first with simple templates and retrieval methods, then with neural encoder-decoder models, later with attention and better training, and finally with large-scale vision-language models. Each stage fixed some of the problems of the one before it, but also left open questions about flexibility, evaluation, and how captions can be adapted for different purposes.

In this thesis we focus on both sides. On the one hand, we build and test classical encoder-decoder models using different image encoders (ResNet, EfficientNet, Vision Transformer), decoders (LSTM and GRU), and attention mechanisms. On the other hand, we experiment with modern vision-language models such as BLIP, BLIP-2, GIT, and LLaVA, paying special attention to how prompts can change the style or purpose of a caption. All experiments are carried out on the Flickr8k dataset, and captions are evaluated not only with standard automatic metrics but also with judgments from a large language model. This way, the thesis compares older and newer approaches in a consistent setting and asks how captioning evolves when moving from task-specific models to more general ones.

Chapter 3

Dataset

In this chapter, we introduce the dataset used in all the experiments in this thesis: **Flickr8k** [6]. It's a common choice for image captioning, and it is used to evaluate all the models we explore later. While it's not a very large dataset, it's still sufficient for all the experiments needed to train CNN-RNN architectures and to test more advanced multimodal vision-language models.

3.1 The Flickr8k Dataset

The dataset contains 8,092 images collected from Flickr, a photo-sharing website. Each image is paired with five human-written captions that describe the same scene in slightly different ways, sometimes focusing on the people, and other times on the background, actions, or objects. This variation enables models to generalize better by learning that images can be described in diverse but semantically consistent ways, a property especially useful for training sequence models like RNNs or attention-based decoders.

The dataset is split into 6,092 images for training, 1,000 for validation, and 1,000 for testing, with no overlap between sets. Each split preserves the five captions per image, ensuring consistent evaluation across models.

Table 3.1: Overview of the Flickr8k dataset

Number of images	8,092
Number of captions	40,460
Captions per image	5

3.2 Caption Characteristics

This section explores some basic properties of the captions in the Flickr8k dataset, such as their length and the most frequently used words. Understanding these patterns helps guide choices in model design and preprocessing.

3.2.1 Length Distribution

Captions in the dataset vary in length, but most are short, descriptive sentences. On average, a caption contains 11.78 words. The shortest caption is a single word, while the longest contains 38 words.

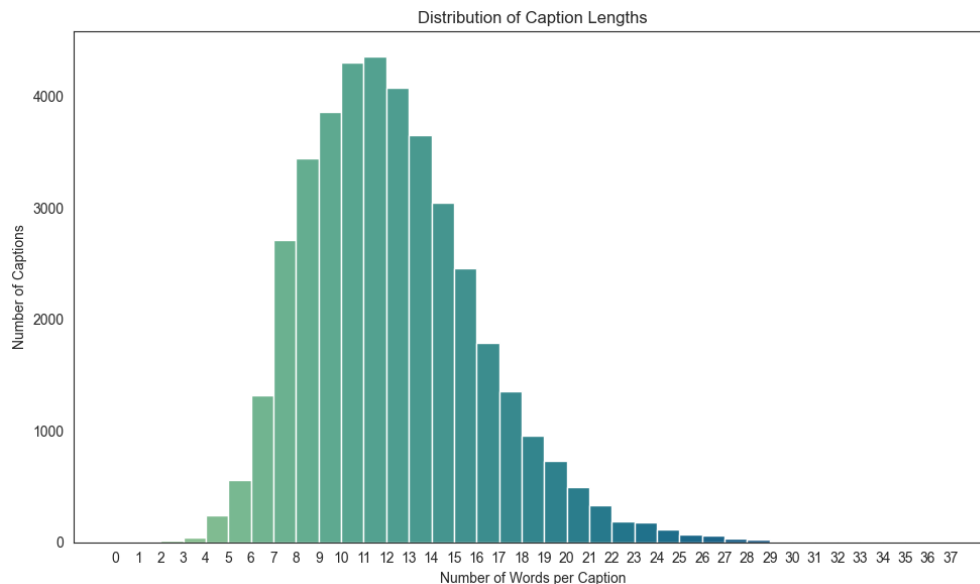


Figure 3.1: Distribution of caption lengths in the Flickr8k dataset.

3.2.2 Vocabulary Characteristics

Table 3.2 gives an overview of the vocabulary before any preprocessing. Across all 40,456 captions in the dataset, there are 476,666 words in total, but only 9,863 are unique. More than 4,000 of those appear just once, which shows how varied the language is. Around a third of the words are stopwords like “the” and “is,” which don’t add much meaning but help make the captions flow naturally.

Table 3.2: Vocabulary statistics before preprocessing

Total words	476,666
Vocabulary size (unique words)	9,863
Rare words (appear only once)	4,296
Stopwords	157,973

3.2.3 Frequent Words

To get a better sense of the caption content, we also examine the most frequently used words, after filtering out punctuation, common stopwords, and other non-informative tokens. As shown in Figure 3.2, the word *dog* appears most often, followed by *man*, *two*, and others. These reflect the kinds of subjects and actions that are typically described in the dataset, which mostly involves people, animals, and activities.

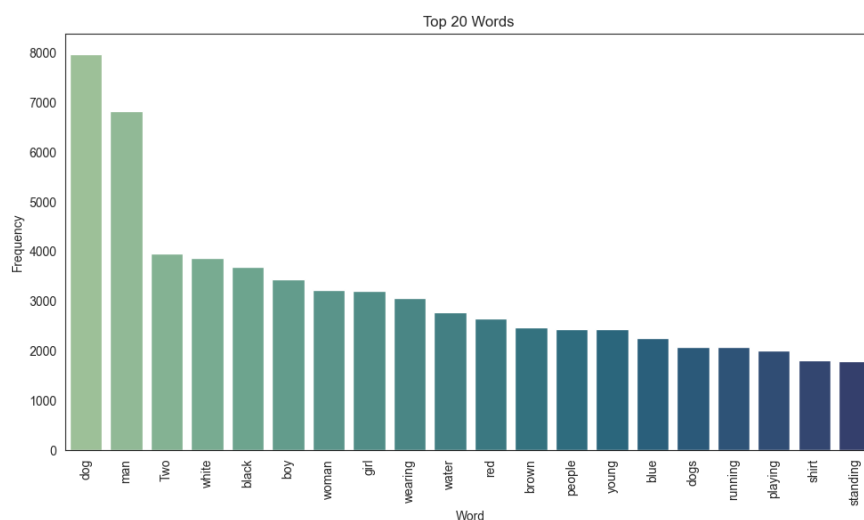


Figure 3.2: Top 20 most frequent words in the Flickr8k dataset (excluding function-words).

3.2.4 Flickr8k Dataset Example

Figure 3.3 displays an example image from the dataset along with the five captions that describe it. Although all the captions refer to the same image, each one focuses on slightly different details. One mentions the tennis ball, another is very short and simply says “dog in air,” while others describe the dog’s color or the grassy background. This variation is important, because it shows that there are many valid ways to describe an image. It also helps the model learn to generate more coherent and natural captions.



1. A dog jumping in midair above grass.
2. A dog jumps up to catch a tennis ball in a grassy field.
3. A golden dog is jumping in an attempt to catch a ball.
4. Dog in air.
5. The dog is in the air above the grass.

Figure 3.3: Example image from Flickr8k.

3.3 Text Preprocessing

Before feeding captions into the model, the text must be cleaned and converted into a format that can be understood by a neural network. This preprocessing step ensures consistency across the dataset and prepares the captions for embedding and training. In this section, we outline the preprocessing pipeline used for the Flickr8k captions, illustrated with the following example: “*A dog jumping in midair above grass.*”

3.3.1 Preprocessing Flow

The preprocessing steps are fairly standard in NLP but are worth discussing them in detail, since many decisions (like whether to keep punctuation or how to handle rare words) can affect the model's performance.

1. **Lowercasing:** All captions are converted to lowercase to ensure that words like Dog and dog are treated as the same token.

The example becomes: a dog jumping in midair above grass.

2. **Punctuation Removal:** Punctuation marks are removed to simplify the vocabulary. In this example, the period at the end is discarded. This helps reduce vocabulary size and noise.

The example becomes: a dog jumping in midair above grass

3. **Tokenization:** Each caption is split into individual words (tokens). We used NLTK's `TreebankWordTokenizer` [44], which handles cases like contractions or punctuation boundaries well.

The example becomes: [a, dog, jumping, in, midair, above, grass]

4. **Special Tokens:** To mark the beginning and end of a caption, two special tokens are added: `<startseq>` and `<endseq>`. These help the decoder understand when to start and stop generating words.

The example becomes: [`<startseq>`, a, dog, jumping, in, midair, above, grass, `<endseq>`]

5. **Vocabulary Construction:** After tokenizing all the captions, a vocabulary is built using the Keras's `Tokenizer` [45]. Each word is assigned a unique integer index. Words that appear only once can optionally be excluded or mapped to an `<unk>` token. For the following experiments, we chose to keep them, as the dataset is relatively small and benefits from maintaining its variability.

6. **Integer Encoding and Padding:** Each caption is then converted into a sequence of integers according to the vocabulary built by the Keras's `Tokenizer` [45]. These indices are not random but correspond directly to the word-index mapping learned from the dataset. Since caption lengths vary, shorter sequences are padded with zeros at the end to match the length of the longest caption in the training set.

The same example after encoding and padding becomes: [3, 1, 17, 22, 9, 31, 27, 4, 5, 0, 0, ..., 0] where 3 = <startseq>, 1 = a, 17 = dog, ..., 5 = <endseq>, and 0 = <pad>.

This preprocessing flow transforms each caption into a clean and fixed-length sequence of integers. These sequences are now preprocessed and suitable for input into the model. Although the numbers themselves do not carry any meaning, they will be later converted into word embeddings. These embeddings provide a semantic representation that help the model understand the relationships and similarities between words.

This preprocessing pipeline forms the foundation for all classical models in this work, ensuring consistent and reproducible input representations. In contrast, the large pretrained vision-language models evaluated later in the thesis operate on raw text without this preprocessing, highlighting a methodological difference between the two approaches.

Chapter 4

Building Captions with Encoder-Decoder Models

In this chapter we work with the encoder-decoder approach, which has been the basis of many image captioning systems. The idea is straightforward: an image is first processed by a visual encoder, usually a CNN or a vision transformer, and then a sequence decoder generates the caption word by word. This approach was introduced in the “Show and Tell” model by Vinyals *et al.* [17] and has since been used in many variations. Here, we use this setup to test different design choices. We compare several image encoders (ResNet50, EfficientNetB0, and ViT), two types of decoders (GRU and LSTM), and a range of attention mechanisms. We also look at how different decoding strategies, such as greedy search and beam search, affect the captions. The goal is to understand how each component of the model affects the caption quality and to establish a strong baseline before exploring more advanced vision-language models.

4.1 Architecture Overview

All the models follow a general encoder-decoder structure, with variations in the encoder (ResNet50, EfficientNetB0, or Vision Transformer), the decoder (GRU or LSTM), and the optional use of an attention mechanism. The encoder, decoder, and attention module are all swappable parts in the experiments. By changing one while keeping the others fixed, we can isolate its effect on caption quality. This makes it easier to deduce, for example, whether performance gains arise from better image features or from the way the decoder uses them.

The encoder-decoder pipeline consists of two main parts: a visual encoder that processes the input image, and a sequence decoder that generates the caption one word at a time. The encoder is typically a pre-trained CNN or a vision transformer that converts the image into a dense feature representation. The decoder is usually an RNN, which uses these features to predict a sequence of tokens.

The output of the encoder is either a global feature vector or a set of spatial features, depending on the architecture. These are passed to the decoder, which uses them to guide word generation. During training, the decoder learns to predict each word in the caption using the previous ground-truth words and the encoded image context. This architecture can be extended with attention mechanisms, such as the one introduced in Show, Attend and Tell, by Xu *et al.* [19], which allow the decoder to focus on specific regions of the image when generating each word. This concept will be explored further in a later section.

Figure 4.1 illustrates the image captioning process, showing how the input image flows through each main component to generate a caption.

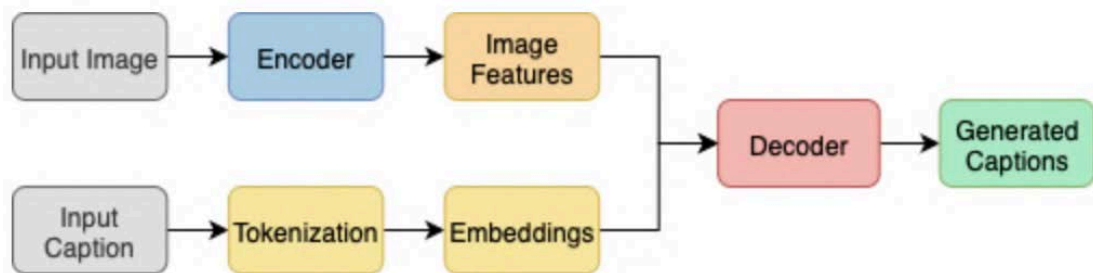


Figure 4.1: General encoder-decoder training pipeline for image captioning, consisting of (1) an encoder for image feature extraction, (2) caption preprocessing to embed the input text, and (3) a decoder that generates the output caption.

4.2 Image Encoders

The encoder is the part of the model that turns an image into features the decoder can work with. The choice of encoder matters a lot, because it controls how much visual detail and structure the model can actually use for captioning. In this thesis, we focus on three types of encoders: ResNet-50, as a widely used CNN baseline, EfficientNetB0, as a smaller but efficient alternative, and the Vision Transformer (ViT), which relies on self-attention rather than convolutions. By comparing them, we can observe not just how well they perform, but also what is gained or lost when moving from traditional CNNs to newer Transformer-based approaches.

4.2.1 ResNet-50

ResNet-50 [46] was included as a strong convolutional baseline due to its proven ImageNet [47] performance and frequent use in captioning literature [1]. With its 50 layers and residual blocks, ResNet-50 can capture higher-level features that help describe complex scenes. Residual connections make it easier to train deep models by learning differences (*residuals*) rather than full transformations.

In classification tasks, ResNet-50 ends with a fully connected layer for class logits. For captioning, this layer is removed, leaving the network as a pure feature extractor. Depending on the model configuration, the output is either:

- A global 2048-dimensional feature vector (via global average pooling) for non-attention models.
- A $7 \times 7 \times 2048$ spatial feature map for attention-based models, flattened into 49 regional descriptors.

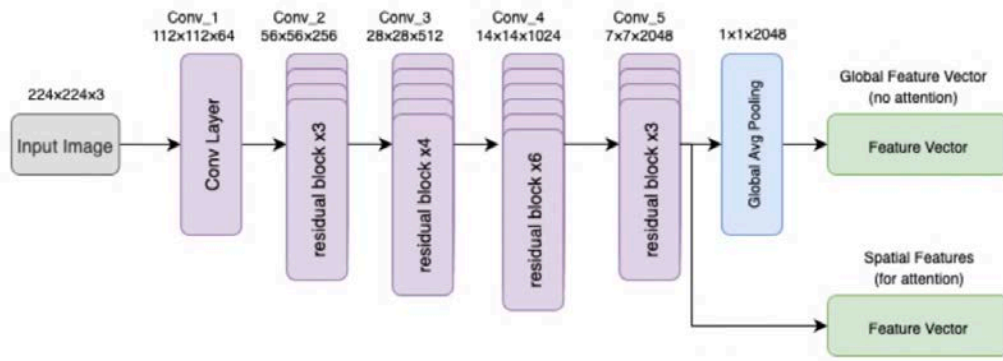


Figure 4.2: ResNet-50 encoder: It outputs a global feature vector or a spatial feature map for attention (figure adapted from [1]).

4.2.2 EfficientNetB0

EfficientNetB0 [48, 49] was chosen to check if a smaller encoder can still produce features good enough for captioning. Many captioning systems must run in real-time or on limited hardware, so understanding the accuracy-efficiency trade-off is critical.

EfficientNetB0 uses *compound scaling* to balance depth, width, and resolution. Its MBConv (Mobile Inverted Bottleneck Convolution) blocks integrate depthwise separable convolutions with squeeze-and-excitation (SE) layers, allowing the network to emphasize important channels with minimal additional computation.

As with ResNet-50, the classification head is removed to yield:

- A 1280-dimensional global vector for non-attention models.
- A $7 \times 7 \times 1280$ feature map for attention-based models.

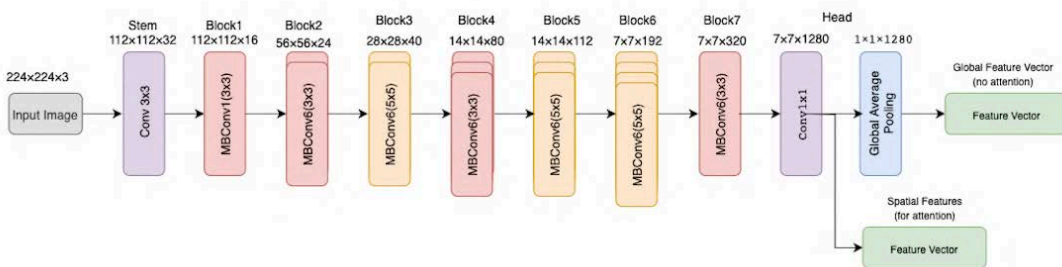


Figure 4.3: EfficientNetB0 encoder: It outputs a global vector or spatial map for attention (figure adapted from [2]).

Comparing EfficientNetB0 to ResNet-50 shows how much captioning performance depends on raw feature quality versus network efficiency.

4.2.3 Vision Transformer

The Vision Transformer (ViT) [3] was included to explore whether a self-attention based vision backbone can offer richer global context than CNNs. Unlike convolutions with fixed receptive fields, ViTs process the image as a sequence of patches, which can directly connect information between distant parts of the image.

Using the pretrained `google/vit-base-patch16-224-in21k` [50], the image is divided into 16×16 patches via a Conv2D layer, producing $14 \times 14 = 196$ patch embeddings of size 768. A special [CLS] token is prepended, and positional embeddings are added before passing the sequence through 12 Transformer encoder layers.

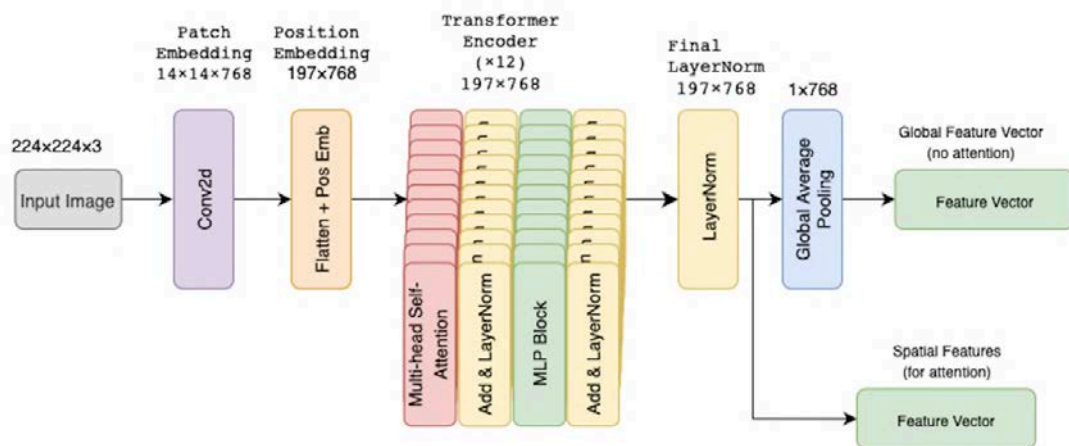


Figure 4.4: ViT encoder: images are split into patches, embedded, positionally encoded, and processed through Transformer layers (figure adapted from [3]).

For non-attention models, the final output is a pooled 768-dimensional vector. For attention-based models, all 197 tokens (patches + [CLS]) are preserved as spatial features. This lets the decoder attend directly to semantically meaningful regions, even if they are far apart in the image.

4.3 Word Embeddings

While the encoder processes images, the decoder must operate in a space, where words are represented numerically, in a way that preserves their meaning. This is achieved through *word embeddings*, which map each token in the caption vocabulary to a dense vector. The choice of embedding type can influence how quickly and effectively the decoder learns language patterns.

In this work, we compare:

- **Randomly initialized embeddings**, learned entirely from scratch during training.
- **Pre-trained GloVe embeddings** [11], initialized from a large text corpus (Wikipedia + Gigaword [51]) and fine-tuned for captioning.

Pre-trained embeddings are expected to offer a head start by providing semantic structure learned from billions of words. If this prior knowledge improves results, it suggests that much of the challenge lies in aligning visual features with existing linguistic concepts rather than learning language from scratch.

Table 4.1: Embedding configurations tested

Embedding Type	Initialization	Trainable	Vector Size
Random (learned)	Random	Yes	256
Pre-trained GloVe	Wikipedia + Gigaword	Yes	200

We tested both embedding configurations with all encoder-decoder models. This way we could determine whether starting from pre-trained GloVe embeddings gave any advantage over learning embeddings from scratch.

4.4 Training Process

Once the encoder produces image features, the decoder must learn to turn them into captions. Training is based on predicting the next word in the sequence, using both the image context and the words seen so far.

The models are trained with teacher forcing, where at each time step the ground-truth word (rather than the model's own prediction) is fed back into the decoder. This approach stabilizes training, prevents early error aggregation, and speeds up convergence.

Formally, during training, the decoder receives two inputs per time step:

1. The embedding of the previous ground-truth word.
2. The image features, either a global vector or an attention-weighted combination of spatial features.

These inputs are fused (via concatenation) and passed into a GRU or LSTM, which maintains a hidden state across time steps. The output is a probability distribution over the vocabulary.

The goal is to minimise the categorical cross-entropy loss between this predicted distribution and the true distribution, where the correct word has probability 1 and all others 0 (one-hot encoding). For a vocabulary of size V , the loss at time step t is:

$$\mathcal{L}_t = - \sum_{i=1}^V y_{t,i} \log(\hat{y}_{t,i})$$

where $y_{t,i}$ is 1 if i is the correct word and 0 otherwise, and $\hat{y}_{t,i}$ is the predicted probability for word i .

When training in batches, captions are padded to the same length for efficiency. However, these `<pad>` tokens are not real words and should not influence the loss. To avoid this, padding masks are applied so that the loss is only computed over valid tokens. Without masking, the model would waste capacity trying to “predict” padding positions, potentially harming both training stability and final caption quality. Figure 4.5 illustrates the training loop.

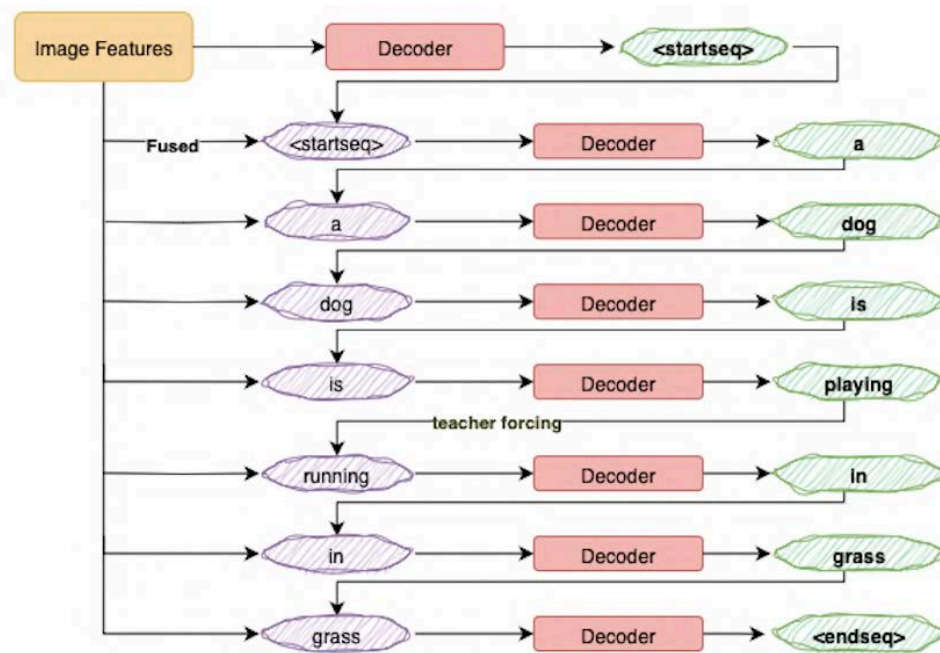


Figure 4.5: Decoder training with teacher forcing: the model conditions on the previous ground-truth word (shown in purple) rather than its own prediction (shown in green).

This setup keeps training the same across all experiments, so any differences in caption quality arise from the model architecture and not from the way the model was trained.

4.5 Attention Mechanisms and Fusion

Using a single global image vector is simple, but it often misses details when multiple objects or actions are present. Attention mechanisms solve this by letting the decoder look at different regions of the image at different time steps [19]. In practice, this means each predicted word can be linked to the parts of the image that matter most for it.

4.5.1 Overall Architecture

In the attention model, the encoder produces a grid of spatial features (e.g., $7 \times 7 \times 2048$ for ResNet-50). Instead of collapsing these into a single vector, the decoder attends to the grid step by step. As shown in Figure 4.6, the decoder state at time t interacts with the spatial features to produce a context vector, which guides the next word prediction.

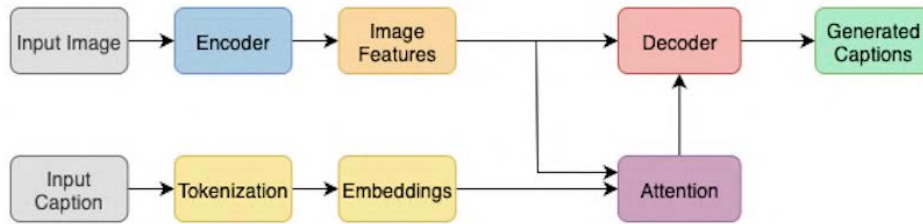


Figure 4.6: Attention in the training pipeline of image captioning. At each step, the decoder can focus on different regions of the feature map.

4.5.2 Computation Process

The attention computation (Figure 4.7) can be summarized in three main steps:

1. **Scoring:** Each encoder hidden state h_i is compared with the current decoder state s_t using a scoring function $f(\cdot)$, producing a set of alignment scores $e_{t,i} = f(s_t, h_i)$. These scores indicate how relevant each encoder position is for generating the next word.
2. **Normalization:** The raw scores are passed through a softmax to obtain the attention weights

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_j \exp(e_{t,j})}.$$

The weights are positive and sum to 1, forming a probability distribution over the encoder states.

3. **Context vector:** Each encoder state h_i is multiplied by its weight $\alpha_{t,i}$, and the weighted states are summed to form the context vector

$$c_t = \sum_i \alpha_{t,i} h_i.$$

This context vector is then combined with the decoder's input (such as the current word embedding x_t) to update the decoder and predict the next word y_t .

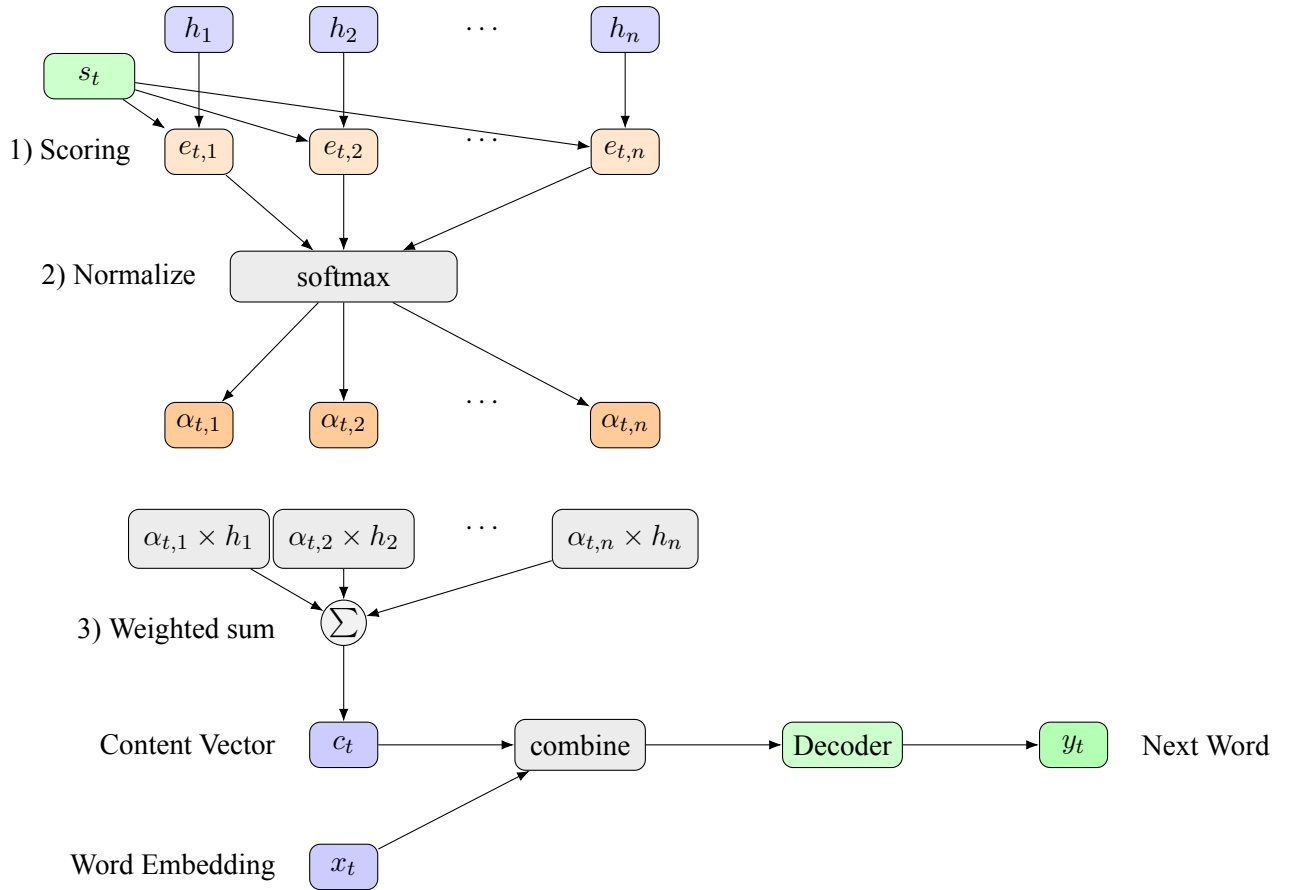


Figure 4.7: Step-by-step computation of attention at decoding step t .

4.5.3 Attention Variants

Two scoring functions were tested:

- **Additive (Bahdanau) Attention** [4]:

$$e_{t,i} = v_a^\top \tanh(W_s s_t + W_h h_i)$$

where W_s , W_h are learned projections and v_a is a learned vector. This approach allows flexible combinations of decoder states and image features.

- **Luong (Multiplicative) Attention** [5]: Given decoder hidden state s_t and encoder features $\{h_i\}_{i=1}^N$, Luong defines several scoring functions:

$$e_{t,i} = \begin{cases} s_t^\top h_i & \text{(dot)} \\ s_t^\top W_a h_i & \text{(general)} \\ (W_s s_t)^\top (W_h h_i) & \text{(concat / projected dot)} \end{cases}$$

where W_a, W_s, W_h are learned projection matrices.

4.5.4 No-Attention (Global Fusion)

To compare with attention, we also trained a baseline without it. In this case, the global image vector is repeated at every step and concatenated with the word embedding. This is a form of late fusion: the image is first encoded by the CNN, and the combination with text happens only during decoding. The approach is computationally simple but does not adjust word by word. Table 4.2 gives an overview.

Table 4.2: Fusion methods tested. Both attention and concatenation are late fusion approaches

Fusion Method	Adaptivity	Fusion Type
Concatenation (no attention)	Static	Late fusion (global vector)
Bahdanau Attention	Dynamic	Late fusion (spatial map)
Luong Attention	Dynamic	Late fusion (spatial map)

4.5.5 Visualizations

Attention can also be visualized by overlaying the weights on the image. Figure 4.8 shows one example: when predicting the word *ball*, the model highlights the ball region, while for *man* it shifts to the person. These visualizations confirm that the model distributes its focus in a way that corresponds to the generated words. The colors represent the strength of attention, with brighter regions indicating higher weights and darker areas indicating little or no focus.



Attention map for “man”.

Attention map for “ball”.

Figure 4.8: Attention maps for selected words (Bahdanau Attention).

4.6 Inference and Decoding

After training, the model must generate captions without access to ground-truth words, a stage known as inference. This is where performance truly matters: while training optimizes the model under teacher forcing, inference exposes it to its own predictions, which can compound errors. Choosing an appropriate decoding strategy is therefore essential, as it directly influences caption quality, diversity, and fluency. At inference time, the encoded image features (either a global vector or a set of spatial vectors) are passed to the decoder along with the special `<startseq>` token. At each time step, the decoder outputs a probability distribution over the vocabulary, from which the next word is selected. This word is fed back into the decoder until an `<endseq>` token is generated or a maximum sequence length is reached. In attention-based models, the attention distribution is recomputed at every step, allowing the model to refocus as the caption unfolds.

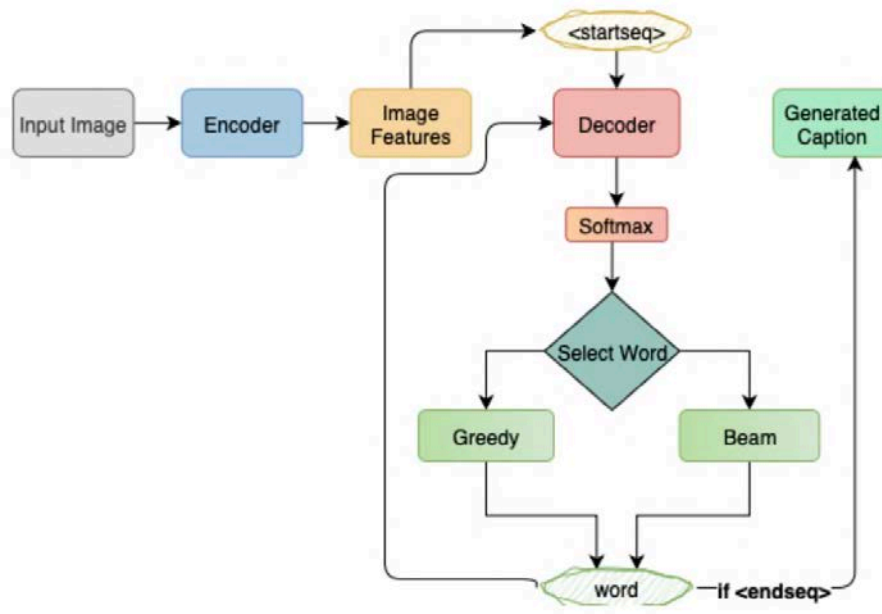


Figure 4.9: Illustration of the inference process.

4.6.1 Decoder Architecture

The decoder is a single-layer LSTM or GRU with a hidden size of 512, followed by a dropout layer ($p = 0.5$) and a softmax output over the vocabulary. Since Flickr8k is relatively small and the goal here is to compare encoders and attention mechanisms, deeper decoders quickly overfit. Both LSTMs and GRUs are capable of modeling long-term dependencies, but GRUs are more lightweight and, in our experiments, gave slightly better results. During inference, the decoder operates autoregressively: each predicted word influences all subsequent predictions. This makes it more sensitive to early mistakes but also gives it flexibility to adapt to the evolving context. This is why decoding strategy selection is also important.

4.6.2 Greedy Decoding

Greedy decoding selects the most probable word at each step. It is simple and fast, making it attractive for real-time applications. However, it only considers local optimality, choosing the highest-probability word at the current time-step, without considering how such choice affects future words. This can lead to suboptimal or repetitive captions.

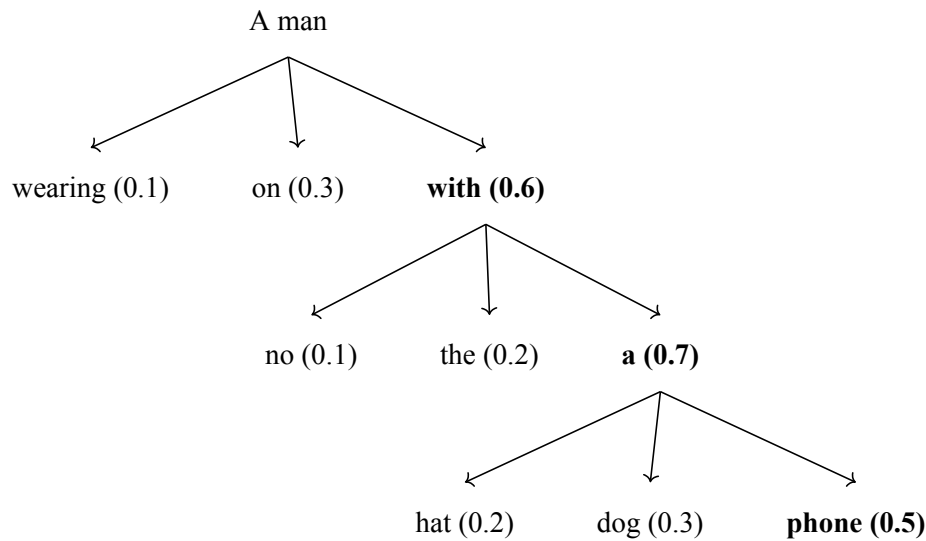


Figure 4.10: Greedy decoding always selects the highest-probability word.

Greedy decoding was included as a baseline to measure the gains from more advanced methods. If it performs close to beam search, the dataset or model may naturally limit the benefits of exploration.

4.6.3 Beam Search Decoding

Beam search minimizes greedy decoding's proximal decision, by maintaining k candidate sequences (beam width) at each step. Instead of committing to a single best word, it explores multiple paths in parallel and keeps those with the highest total probability.

This strategy often generates more coherent and descriptive captions, especially for longer or more complex scenes. It comes at the cost of higher computation, but the trade-off can be worthwhile.

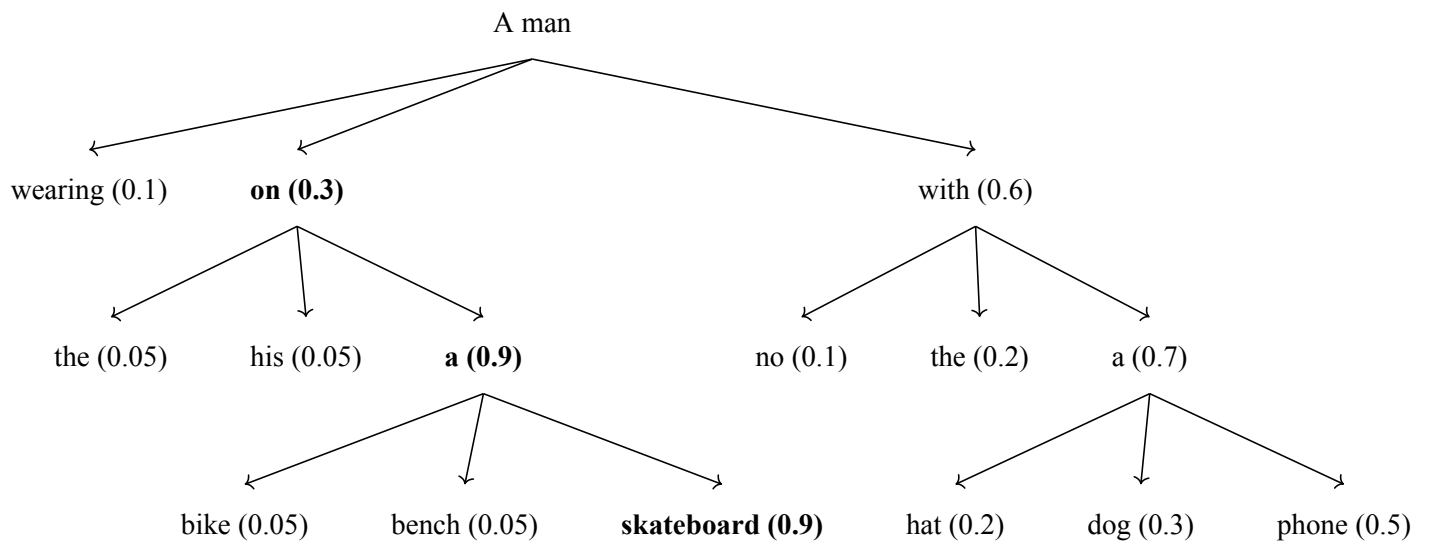


Figure 4.11: Beam search explores multiple candidates and chooses the best overall sequence.

Beam search was included to quantify the benefits of broader search. In practice, small beam widths ($k = 3$ or $k = 5$) often strike a balance between quality and efficiency. Larger beams can improve accuracy further but risk producing overly generic captions due to probability averaging.

Chapter 5

Better Captions with Vision-Language Models

The previous chapter covered classical encoder-decoder models, which generate a fixed caption for each image and cannot adapt style, detail, or purpose. In contrast, modern vision-language models (VLMs) can produce captions in many styles (short, long, descriptive, creative, or accessibility-focused) and some can even answer questions about the image. This flexibility comes from *promptability*, guiding caption generation at inference time with natural language instructions.

Even though they still follow an encoder-decoder setup, VLMs are built very differently from the CNN-RNN systems in Chapter 4. Instead of training small models from scratch on Flickr8k, they use large pretrained vision encoders, often Transformers, together with language models trained on huge collections of image-text pairs. This makes them capable of zero-shot captioning, and when they are promptable, they can follow instructions without extra training.

This chapter presents the four VLMs evaluated in this work, BLIP [7], GIT [8], BLIP-2 [9], and LLaVA [10], ranging from non-promptable captioners to fully promptable, and compares their caption quality and flexibility.

5.1 Models Overview

The models differ in four main aspects: whether they are promptable, their primary use-case, the architecture of their vision and language components, and their parameter count. Table 5.1 summarizes these differences.

Table 5.1: Overview of the vision-language models used in this study

Model	Promptable	Used For	Parameters
BLIP	No	Captioning	~213M
GIT (large)	No	Captioning	~345M
BLIP-2 (OPT 2.7B)	No	Captioning	~2.7B
LLaVA (Vicuna-7B)	Yes	Captioning + Visual Question Answering	~7B

The first three models, BLIP, GIT, and BLIP-2, act more like traditional captioners, producing a fixed output for a given image. LLaVA stands apart as a promptable model capable of tailoring captions to specific instructions. Together, they allow us to measure both pure captioning ability and adaptability to user intention.

5.1.1 Bootstrapped Language-Image Pretraining (BLIP)

BLIP was among the first models to significantly outperform earlier CNN-RNN captioners. Its training strategy addresses the fact that large-scale image-text datasets often contain noisy or imperfect captions. BLIP uses a dual-objective approach: one head performs image-text matching to filter or reweight training pairs, and the other head generates the captions. This allows the model to learn the important associations between images and text, even when the data is imperfect.

Architecturally, BLIP has a ViT encoder for images and a Transformer-based decoder for text. The image-text matching head is only used during training, while at inference the model generates captions directly from the visual features.

In this thesis, we use the BLIP-base model (213M parameters). It is *not* promptable, meaning that for a given image it will always produce the same caption. BLIP is included as a

representative of a strong, pre-trained, but fixed-output captioner, providing a baseline for comparing against both older encoder-decoder systems and more advanced VLMs.

5.1.2 Generative Image-to-Text Transformer (GIT)

GIT follows a clean, minimal design: a Vision Transformer encodes the image, and a Transformer-based decoder generates the caption. Unlike BLIP, GIT does not use a separate matching objective or an external large language model. Its decoder is trained from scratch purely for the image-to-text task, making it fully self-contained.

The GIT-large model (345M parameters) is used in the experiments. Like BLIP, it is non-promptable, returning a fixed caption for each image. The inclusion of GIT allows us to test whether an efficient architecture, trained directly for caption generation, can match or exceed the performance of BLIP’s multi-objective approach.

5.1.3 BLIP-2

BLIP-2 extends the BLIP framework by connecting a frozen image encoder (e.g., ViT) to a large pre-trained language model (e.g., OPT or Vicuna) via a small, trainable *query transformer*. This connector module learns to translate visual features into a form the language model can understand.

The efficiency of BLIP-2 comes from training only the connector, while keeping both the vision and language components frozen. This makes it possible to leverage extremely large language models without provoking the cost of full fine-tuning. In the experiments, we use BLIP-2 with the OPT-2.7B language model.

While BLIP-2 inherits the scale and linguistic fluency of its language model, it remains non-promptable in the sense used here: its caption is determined solely by the image input, with no scope for varied outputs via instructions. BLIP-2’s role in this study is to test whether connecting to a large language model improves caption quality even without promptability.

5.1.4 Large Language and Vision Assistant (LLaVA)

LLaVA combines CLIP’s image encoder [52] with the Vicuna-7B language model [53], enabling both captioning and visual question answering (VQA). What distinguishes LLaVA from the other models here is its *instruction-following* ability: captions can be tailored by providing a natural language *prompt* at inference time.

For example, LLaVA can generate a small description, a list of detected objects, produce an accessibility-focused caption, or create a humorous one, all from the same image, simply by changing the input prompt. This flexibility makes it useful in contexts where captions need to serve different purposes.

In the experiments, we evaluate LLaVA on the Flickr8k test set using six different prompt types, ranging from short factual descriptions to longer, stylistically varied captions. This approach allows to test not only just the caption quality but also the model’s adaptability to diverse user instructions.

Chapter 6

Evaluation Metrics

Evaluating image captioning models is inherently subjective. Unlike classification tasks with a single correct label, many different captions can validly describe the same image. A model may generate a sentence that uses different words from the references, yet be still perfectly accurate. As a result, traditional metrics such as accuracy or precision are insufficient.

To address this, the field uses metrics that compare generated captions to multiple human-written references, assessing similarity in terms of words, phrases, and structure, and in some cases, meaning. In this work, we use the most widely adopted metrics in captioning: **BLEU-1** to **BLEU-4**, **METEOR**, **ROUGE-L**, and **CIDEr**, along with embedding-based methods **BERTScore** and **CLIPScore**. Each captures a different aspect of caption quality, making them complementary when used together.

To explain the behaviour of different metrics, consider the following example:

- **Reference:** *“A black and white dog is running in the park.”*
- **Prediction:** *“A black dog runs in the park.”*

The prediction is semantically similar but excludes words (*and white*) and uses different verb forms (*runs* vs. *is running*).

6.1 BLEU: Bilingual Evaluation Understudy

BLEU [39] measures the proportion of n -grams (contiguous word sequences) in the generated caption that also appear in the references, using *modified precision* and a *brevity penalty* (BP) to discourage very short outputs:

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^N w_n \cdot \log p_n \right)$$

Here, p_n is the modified precision for n -grams, w_n is their weight (e.g., $1/4$ for BLEU-4), and BP is:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

where c and r are candidate and reference lengths.

Example effect: For our example, BLEU evaluates different n -gram levels:

Reference n-grams:

- Unigrams: {A, black, and, white, dog, is, running, in, the, park}
- Bigrams: {A black, black and, and white, white dog, dog is, is running, running in, in the, the park}
- Trigrams: {A black and, black and white, and white dog, white dog is, dog is running, is running in, running in the, in the park}
- 4-grams: {A black and white, black and white dog, and white dog is, white dog is running, dog is running in, is running in the, running in the park}

Prediction n-grams:

- Unigrams: {A, black, dog, runs, in, the, park}
- Bigrams: {A black, black dog, dog runs, runs in, in the, the park}
- Trigrams: {A black dog, black dog runs, dog runs in, runs in the, in the park}
- 4-grams: {A black dog runs, black dog runs in, dog runs in the, runs in the park}

Matches:

- Unigrams: $p_1 = 6/7 = 0.86$ (A, black, dog, in, the, park)

- Bigrams: $p_2 = 3/6 = 0.50$ (*A black, in the, the park*)
- Trigrams: $p_3 = 1/5 = 0.2$ (*in the park*)
- 4-grams: $p_4 = 0$

The candidate length is $c = 7$, while the reference length is $r = 10$, so the brevity penalty BP is:

$$\text{BP} = e^{1-r/c} = e^{1-10/7} = e^{-3/7}.$$

So the BLEU scores become:

$$\text{BLEU-1} = e^{-3/7} \cdot \frac{6}{7} = 0.56,$$

$$\text{BLEU-2} = e^{-3/7} \cdot \sqrt{\frac{6}{7} \cdot \frac{1}{2}} = e^{-3/7} \cdot \sqrt{\frac{3}{7}} = 0.42,$$

$$\text{BLEU-3} = e^{-3/7} \cdot \left(\frac{6}{7} \cdot \frac{1}{2} \cdot \frac{1}{5} \right)^{1/3} = e^{-3/7} \cdot \left(\frac{3}{35} \right)^{1/3} = 0.28,$$

$$\text{BLEU-4} = e^{-3/7} \cdot \left(\frac{6}{7} \cdot \frac{1}{2} \cdot \frac{1}{5} \cdot 0 \right)^{1/4} = 0.$$

These results show how BLEU quickly decreases as n increases. While the candidate caption is semantically correct, BLEU-4 collapses to zero due to missing exact 4-gram matches. This illustrates BLEU's sensitivity to lexical variation and its limitation in capturing semantic similarity.

6.2 METEOR: Metric for Evaluation of Translation with Explicit ORdering

METEOR [41] improves on BLEU by incorporating:

- Word stemming (“runs” vs. “running”),
- Synonym matches (“bike” vs. “bicycle”), and
- Word order, by penalizing fragmented matches.

The score is based on a harmonic mean of unigram precision (P) and recall (R), with recall weighted higher:

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P}.$$

To capture word order, METEOR introduces a fragmentation penalty:

$$\text{Penalty} = 0.5 \cdot \left(\frac{\text{chunks}}{\text{matches}} \right)^3,$$

where *matches* are all aligned words (exact, stem, synonyms), and *chunks* are sequences of consecutive matches that appear in the same order in both sentences. The final score is:

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - \text{Penalty}).$$

Example effect: METEOR aligns the following words: *A/A, black/black, dog/dog, runs/running* (stem match), *in/in, the/the, park/park*. The words *and, white, and is* have no matches, giving 7 matches in total.

- Precision: $P = 7/7 = 1.0$ (all candidate words matched).
- Recall: $R = 7/10 = 0.70$ (7 of 10 reference words matched).
- $F_{\text{mean}} = \frac{10 \cdot 1.0 \cdot 0.70}{0.70 + 9 \cdot 1.0} = 0.72$.

Now, consider how the matches are grouped. We obtain three chunks: *A black dog, runs, in the park*.

Because the matches are not one continuous sequence, the alignment is fragmented. The penalty is:

$$\text{Penalty} = 0.5 \cdot \left(\frac{3}{7} \right)^3 = 0.03.$$

Thus, the final METEOR score is:

$$\text{METEOR} = 0.72 \cdot (1 - 0.03) = 0.69.$$

This example shows how METEOR credits synonyms and stems (“runs” vs. “running”), while penalizing discontinuities in word order via the chunk-based penalty, resulting in a higher and more linguistically accurate score than BLEU.

6.3 CIDEr: Consensus-Based Evaluation

CIDEr [24] rewards captions that use *key* phrases that appear consistently across references, weighting *n*-grams by TF-IDF [54, 55]:

$$\text{CIDEr}_n = \frac{1}{M} \sum_{i=1}^M \frac{g_n(c) \cdot g_n(s_i)}{\|g_n(c)\| \cdot \|g_n(s_i)\|}$$

where M is the number of references and $g_n(\cdot)$ is the TF-IDF vector over all possible n -grams in the dataset vocabulary. Each n -gram is weighted by its Term Frequency in the caption and Inverse Document Frequency across the dataset.

Example effect: Key n -grams like *black dog*, *running*, and *park* have high TF-IDF weights as distinctive descriptors. The semantic overlap between *runs/running* and shared content words results in a CIDEr score of 0.58, higher than BLEU because it rewards conceptual similarity through TF-IDF weighting rather than exact matches.

6.4 ROUGE-L: Longest Common Subsequence

ROUGE-L [40] focuses on sequence-level similarity by measuring the *longest common subsequence* (LCS) between candidate and reference:

$$\text{ROUGE-L} = \frac{\text{LCS}(c, r)}{\text{length of } r}$$

It gives partial credit for in-order matches even when exact n -gram boundaries differ, making it useful for evaluating sentence fluency.

Example effect: The longest common subsequence between the reference and prediction is “*A black dog ... in the park*” with length 6. This gives $\text{ROUGE-L} = 6/10 = 0.60$, capturing the structural similarity despite the missing words and verb form changes, unlike BLEU.

6.5 Embedding-Based Metrics: BERTScore and CLIPScore

Word-overlap metrics can penalize correct captions that use different vocabulary or structure. To address this, the newer approaches evaluate meaning in a continuous embedding space.

6.5.1 BERTScore

BERTScore [12] uses contextual embeddings from a pretrained language model (e.g., BERT [27], RoBERTa [56]) to align semantically similar words between candidate and reference. The similarity between a candidate token embedding x_i and a reference token embedding y_j is computed using the cosine similarity:

$$\cos(x_i, y_j) = \frac{x_i \cdot y_j}{\|x_i\| \|y_j\|}.$$

Precision and recall are then defined as the average maximum similarity for each token:

$$\text{Precision} = \frac{1}{|c|} \sum_{i \in c} \max_{j \in r} \cos(x_i, y_j), \quad \text{Recall} = \frac{1}{|r|} \sum_{j \in r} \max_{i \in c} \cos(y_j, x_i),$$

where c and r denote the sets of tokens in the candidate and reference captions.

Finally, BERTScore reports the harmonic mean of precision and recall:

$$\text{BERTScore-F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Example effect: In our running example, BERTScore aligns semantically similar words: *runs* matches with *running* (high cosine similarity), while identical words such as *dog*, *park*, and *black* receive perfect scores (very close to 1.0). Because the contextual embeddings capture that both sentences describe the same scene, the resulting BERTScore-F1 is 0.89, indicating strong semantic similarity despite some lexical differences.

6.5.2 CLIPScore

CLIPScore [13] bypasses references entirely by measuring the cosine similarity between the image embedding and the caption embedding in CLIP’s joint vision-language space [52]:

$$\text{CLIPScore} = \cos(\text{CLIP}(\text{image}), \text{CLIP}(\text{caption}))$$

This is especially valuable for evaluating prompt-based captions, where there may be no matching human reference but the description is still visually accurate.

Example effect: If the image shows a black dog running in a park, both the reference and prediction describe the same visual scene accurately. The key visual elements (black dog, park, running motion) are preserved in both descriptions. CLIPScore would be high, showing high visual accuracy despite the missing descriptive words.

6.6 Why all these Metrics

Each of the metrics used in this thesis captures a different side of caption quality. BLEU and ROUGE-L look at exact word and phrase matches, making results easy to compare with references. METEOR goes further by recognising synonyms and word variations, giving scores that match human judgment more closely. CIDEr rewards distinctive and informative phrases

found in the references. BERTScore measures how close captions are in meaning, using embeddings instead of exact matches. CLIPScore checks how well a caption matches the image directly, without needing references, useful for promptable models that can produce many different, but still accurate outputs. Using them together makes it possible to judge both the precision of fixed-caption systems and the flexibility of prompt-based models. Still, even these measures can fall short for highly adaptable VLMs, which is why more recent work has started exploring evaluation with Large Language Models as judges.

Chapter 7

Experiments and Results

This chapter presents the experimental results of our image captioning models. We begin with classical encoder-decoder architectures and then move to large pretrained vision-language models (VLMs), including promptable ones. The focus is not only on reporting scores, but also on how different design choices influence the captions that are generated. As we progress from classical models to VLMs and then to prompt-driven approaches, evaluation itself becomes more complex. Classical models can be assessed with standard metrics, but VLMs introduce flexibility and variation that make evaluation more subjective and raise new challenges.

7.1 Evaluation of Classical Encoder-Decoder Architectures

The encoder-decoder pipeline has long served as the foundation for image captioning. In this section, we explore how different configurations affect the quality of generated captions. All models were trained on the Flickr8k dataset under identical preprocessing and evaluation settings.

7.1.1 Comparison of Image Encoders

We began by isolating the effect of the visual encoder. Three popular choices were tested: ResNet50, EfficientNetB0, and the Vision Transformer (ViT). All were paired with additive attention, GRU decoder, GloVe embeddings, and beam search.

Table 7.1: Performance of image encoders with identical downstream setup

Encoder	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
ViT	0.59	0.39	0.26	0.18	0.40	0.54
EffNetB0	0.55	0.36	0.24	0.16	0.37	0.46
ResNet50	0.56	0.36	0.24	0.16	0.37	0.44

ViT gave the best results across all metrics. This can be explained by its use of self-attention, which allows the model to link information from distant parts of the image and build a more global view. ResNet and EfficientNet also performed well, but they are based on convolutional filters that focus on local neighborhoods. As a result, they may miss some of the wider scene structure, such as how different objects relate to each other.



Reference (Flickr8k): Five people stand on the beach looking out at the ocean.

ViT: Five men are walking on the beach looking at the ocean.

EffNetB0: Three people are walking along a beach.

ResNet50: Two men stand on the beach looking at the ocean.

Figure 7.1: Image used to compare captions generated by different visual encoders.

Figure 7.1 shows an example of these differences. The reference caption describes five people on the beach. The ViT-based model comes closest, correctly identifying the group and the ocean in the background. EfficientNet underestimates the number of people, while ResNet reduces the group even further to two. This suggests that CNN-based encoders may lose some of the broader scene context, while ViT maintains it more effectively.

7.1.2 Attention and Fusion Strategies

Next, we tested how image features can be combined with the text sequence. Using ViT as the encoder, we compared three options: concatenation, additive attention, and Luong attention. The results are shown in Table 7.2. The aim was to check if attention helps beyond a simple global fusion.

Table 7.2: Performance of attention and fusion strategies (ViT encoder)

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
Concatenation	0.54	0.35	0.23	0.15	0.36	0.44
Additive Attention	0.59	0.39	0.26	0.18	0.40	0.56
Luong Attention	0.56	0.38	0.26	0.17	0.36	0.54

Table 7.2 shows that additive attention achieved the best scores across all metrics. Concatenation remained close but lacked the ability to adapt its focus. Luong attention was competitive but slightly less consistent.

A qualitative example is given in Figure 7.2. Concatenation mistakes the bike for a motorcycle. Additive attention describes the scene correctly. Luong attention captures the racing context by including both riders. This confirms that attention helps to correct errors and add detail.



Reference (Flickr8k): A man in a red helmet jumps with his bike on a dirt track.

Concatenation: A man is riding a motorcycle.

Additive attention: A man in a red helmet is riding a bike on a dirt track.

Luong attention: Two riders are racing on a bumpy track.

Figure 7.2: Captions generated by different fusion strategies.

To see how this focus changes word by word, Figures 7.3 and 7.4 show attention maps. The colors represent the strength of attention, with brighter regions indicating higher weights and darker areas indicating little or no focus. Additive attention spreads weights more broadly, often including some background, while Luong attention creates sharper, localized peaks. This means that additive attention captures context more softly, while the Luong one enforces a tighter link between words and regions.

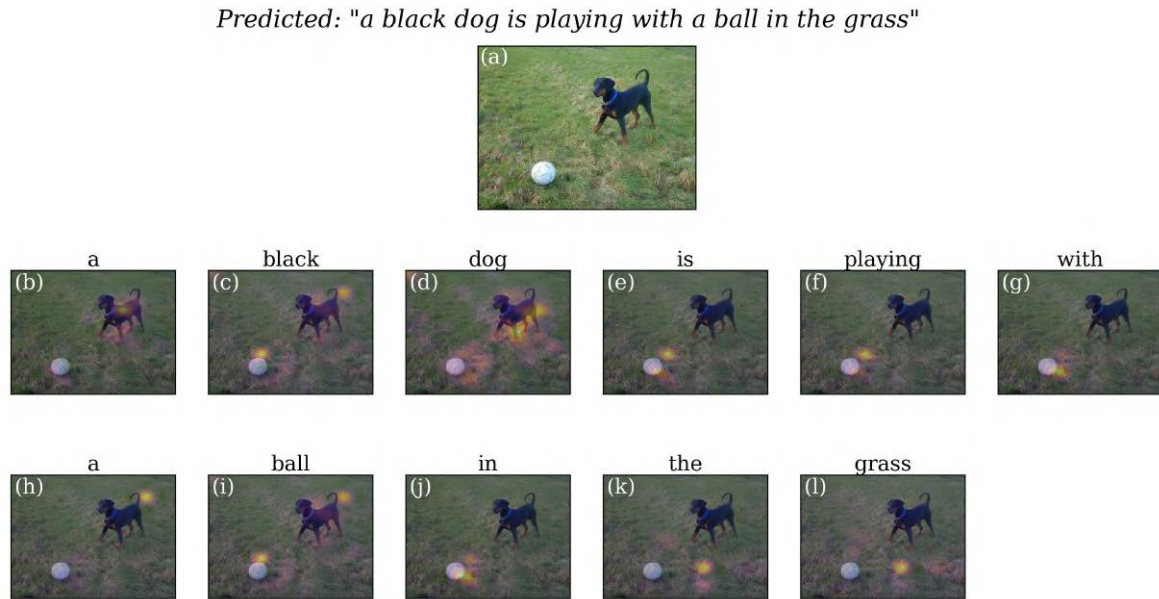


Figure 7.3: Word-by-word visualization of additive attention using ViT features.

Figure 7.3 shows how additive attention works during caption generation. The original image is shown in (a), while (b)-(l) display the attention maps for each predicted word. For object words like “dog” in (d), the model focuses on the dog. For “ball” in (i), attention is split between the ball and the dog. The weights are higher on the ball but still significant on the dog, showing that the model considers both objects together when generating the word. In contrast, function words such as “is” (e) or “the” (k) show more diffuse attention, often spread across the background. At each decoding step, the model shifts focus to different regions, linking objects and context as the caption is formed. More examples, including both successful and failure cases, are provided in Appendix A.1.

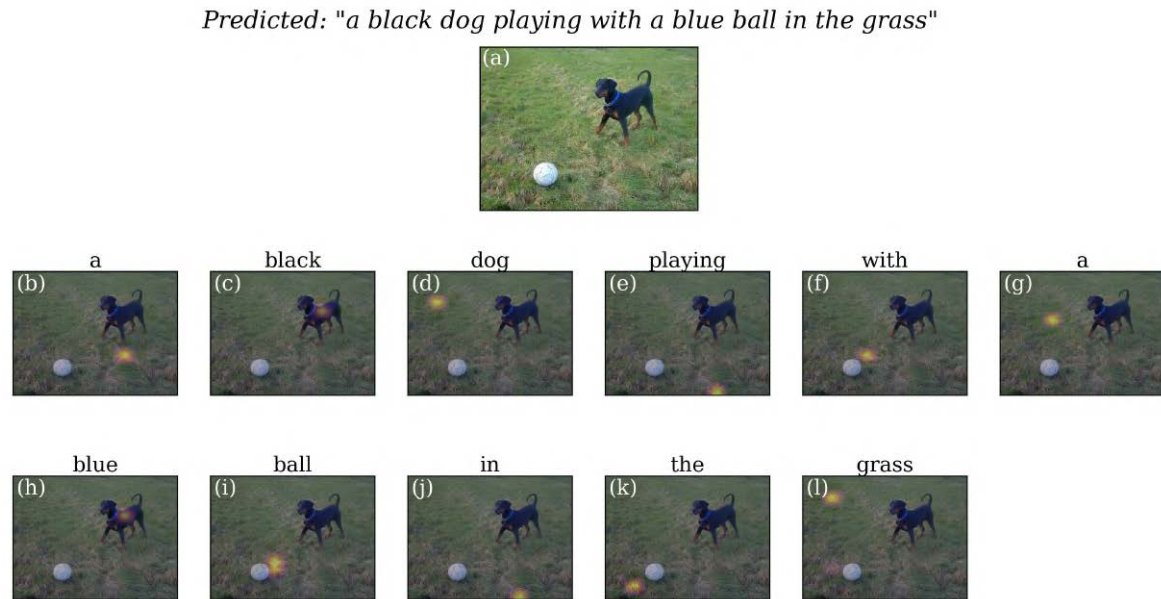


Figure 7.4: Word-by-word visualization of Luong attention using ViT features.

Figure 7.4 shows the same example with Luong attention. Here, the model still attends to the dog for “dog” (d) and to the ball for “ball” (i), but the focus is sharper and more localized than with additive attention. Additive tends to spread weights more broadly, sometimes including background, while Luong enforces a tighter alignment. This suggests that additive provides softer, context-aware focus, whereas Luong attention highlights specific regions more directly.

7.1.3 Training Dynamics

To ensure that the models trained in a stable and well-converged manner, we tracked the evolution of training and validation loss and accuracy across epochs. Figure 7.5 shows the curves for the ViT encoder with additive attention and GloVe embeddings, our best-performing configuration among the classical models, and Figure 7.6 shows the curves for the same model, with Luong attention for the comparison.

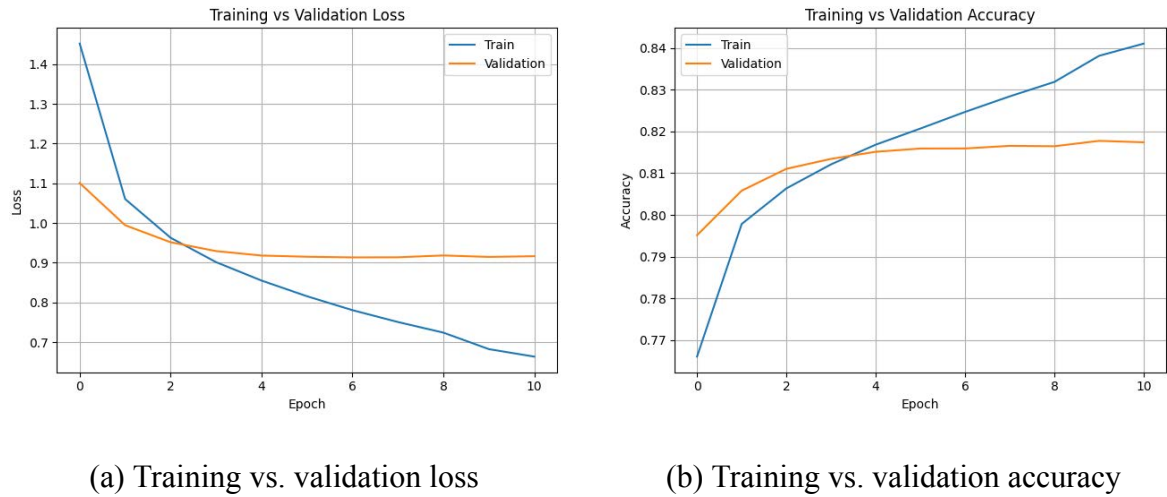


Figure 7.5: Training dynamics of the ViT + Additive Attention + GloVe model. Validation loss stabilizes around epoch 3 or 4 while accuracy plateaus shortly afterwards, indicating convergence without overfitting.

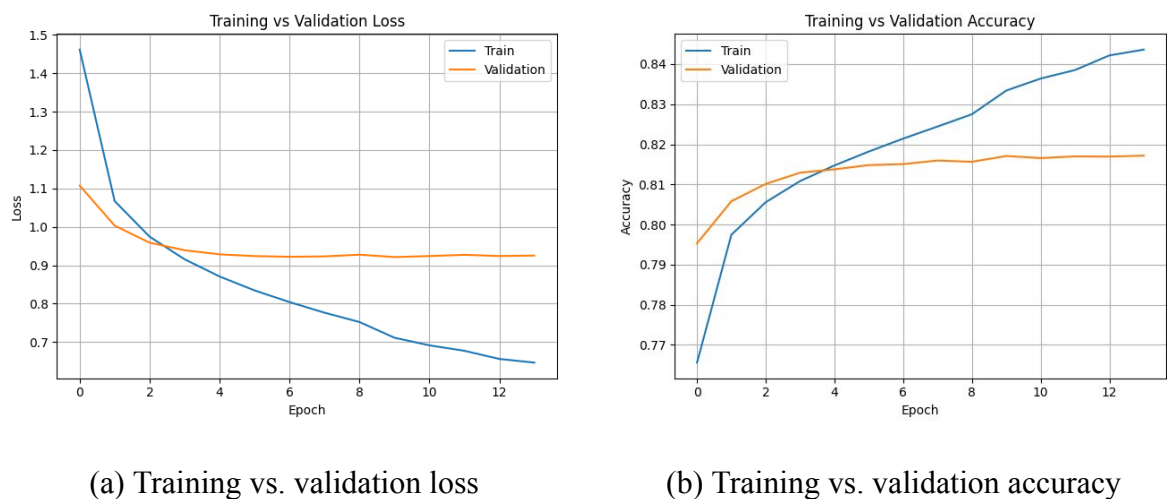


Figure 7.6: Training dynamics of the ViT + Luong Attention + GloVe model. Validation loss stabilizes again around epoch 3 or 4 while accuracy plateaus shortly afterwards, indicating convergence (slower than additive attention) without overfitting.

7.1.4 Word Embeddings: Trainable vs. Pretrained

With the visual encoder and attention mechanism in place, we turned our focus to the language component of the model. Specifically, we examined how different word embedding strategies affect caption generation. Word embeddings are the model's first step in processing language, transforming words into numerical vectors that can be used during decoding.

We compared two widely used approaches: (1) trainable embeddings, which are randomly initialized and updated during training, and (2) pretrained GloVe embeddings, which are learned from large external text datasets and kept fixed during training.

Table 7.3: Performance of embedding types (ViT + Additive Attention + GRU + Greedy Decoding)

Embedding Type	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
Trainable Embeddings	0.569	0.385	0.261	0.175	0.408	0.543
GloVe (Pretrained)	0.580	0.387	0.255	0.167	0.400	0.535

As shown in Table 7.3, the two methods achieved similar performance overall. GloVe embeddings slightly outperformed trainable embeddings in BLEU-1 and BLEU-2, which focus on shorter n -gram overlaps. In contrast, the trainable embeddings performed slightly better on BLEU-4, METEOR, and CIDEr, metrics that reward more complete and semantically accurate captions. This suggests that while GloVe embeddings provide strong general linguistic priors, trainable embeddings can better adapt to the specific characteristics and vocabulary of the dataset.

To better understand the influence of embedding choice, we compared the generated captions on a specific example, as shown below in Figure 7.7.



Reference (Flickr8k): The dog is running along the beach next to the ocean with a stick in its mouth.

Trainable Embeddings: A dog is running near the water.

GloVe Embeddings: A dog is running through the water with a stick in its mouth.

Figure 7.7: Image used to compare the effect of embedding type on generated captions.

In this case, only the GloVe-based model correctly included the word “stick” in the caption. The version using trainable embeddings ignored it entirely. This difference may be explained by word frequency: “stick” is relatively rare in the Flickr8k dataset. As a result, a model learning embeddings from scratch may not have encountered the word often enough to generate it confidently. In contrast, GloVe embeddings are trained on large-scale text datasets and already encode semantic information about the word “stick” and its common associations (e.g., with dogs, play, and outdoor scenes).

7.1.5 Decoder and Search Strategies

Finally, we looked at the last part of the captioning pipeline: the decoder and how it generates sentences word by word. GRUs slightly edged out LSTMs in this dataset, and beam search consistently produced better-formed captions compared to greedy decoding. The results are shown in Table 7.4 and Table 7.5.

Table 7.4: Effect of decoding strategy (ViT + GRU + Additive Attention)

Decoding Strategy	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
Greedy (Beam = 1)	0.52	0.34	0.22	0.15	0.38	0.45
Beam Search (k = 3)	0.59	0.40	0.27	0.19	0.41	0.57
Beam Search (k = 5)	0.57	0.41	0.28	0.18	0.42	0.56

Table 7.5: LSTM vs. GRU decoder comparison (ViT + Additive Attention)

Decoder	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
LSTM	0.55	0.37	0.24	0.16	0.39	0.50
GRU	0.57	0.38	0.25	0.17	0.39	0.52



Figure 7.8: Caption outputs for different beam widths on the same image.

Reference (Flickr8k): A child in a blue uniform kicks a soccer ball.

Greedy (Beam = 1): A young boy in a blue shirt is jumping in the air.

Beam Search (k = 3): A young boy in a blue shirt is playing soccer.

Beam Search (k = 5): A young boy wearing a blue shirt is playing with a soccer ball.

Beam Search (k = 7): A young boy in a blue shirt kicks a soccer ball during a soccer game.

Beam Search (k = 10): A young boy kicks a soccer ball during a soccer game.

Table 7.6: Decoding time per caption for different beam widths (example of Figure 7.8)

Beam Width	Predicted Caption	Time (s)
1 (Greedy)	A young boy in a blue shirt is jumping in the air.	0.33
3	A young boy in a blue shirt is playing soccer.	2.39
5	A young boy wearing a blue shirt is playing with a soccer ball.	2.98
7	A young boy in a blue shirt kicks a soccer ball during a soccer game.	8.90
10	A young boy kicks a soccer ball during a soccer game.	16.23

As shown in Figure 7.8, greedy decoding produces a plausible caption but omits the soccer context entirely. As the beam width increases, captions become more accurate and descriptive, correctly identifying the ball and the game. Table 7.6 illustrates the efficiency trade-off: while beam search with $k = 3\text{--}5$ improves quality at modest cost, wider beams ($k \geq 7$) yield only minor gains but dramatically increase decoding time. This suggests that in practice, a beam width of 3–5 provides the best balance between accuracy and efficiency.

These experiments showed what can be achieved with classical models when every part of the system is trained from scratch on Flickr8k. We designed the encoder, attention, and decoder ourselves, and the results reflect the limits of learning directly from a small dataset. The next step is different: rather than training everything, we now test models that have already been pretrained on massive collections of images and text. This shifts the focus from learning on Flickr8k to evaluating how well such models can adapt and generalize to it without additional training. It also raises a new question: how should these systems be evaluated, given that their outputs are often more flexible and harder to judge with standard metrics alone?

7.2 Evaluation of Pretrained Vision-Language Models

Unlike the custom-built architectures we trained earlier, pretrained vision-language models arrive already equipped with knowledge. They have been trained on millions of image-text pairs, exposed to a wide range of styles, and optimized for general-purpose understanding. In this section, we evaluate three such models, BLIP-1, BLIP-2, and GIT, on 1,000 Flickr8k test images in a fully zero-shot setting, meaning they were not fine-tuned on this dataset in any way. The goal is simple: to see how well these models perform without any task-specific fine-tuning, and whether they can outperform models trained specifically for the task.

In addition to classical metrics like BLEU and CIDEr, we also include two embedding-based scores, BERTScore and CLIPScore, which evaluate semantic similarity and vision-language alignment respectively. These are particularly useful for judging the outputs of large-scale pretrained models that may phrase things differently from the reference captions.

Table 7.7: Zero-shot performance of pretrained VLMs

Model	BLEU-1	BLEU-4	METEOR	CIDEr	BERTscore	CLIPscore
BLIP-1	0.7610	0.2040	0.3926	0.5976	0.9090	0.3309
GIT	0.7456	0.3192	0.5091	0.9503	0.9383	0.3613
BLIP-2	0.7872	0.2470	0.4885	0.8746	0.9199	0.3409



Figure 7.9: Image used to compare captioning styles of BLIP, BLIP-2, and GIT.

Reference (Flickr8k): There are two kids, a soccer ball, a dog and a goat outside in a field.

BLIP: A group of people playing with a dog in a field.

BLIP-2: A group of people playing soccer with a goat.

GIT: A group of people playing soccer with a goat and a goat.

ViT+Attention+GRU: Two children are playing with a dog in a field.

Each model gives a different caption for the same image, depending on how it was trained and how it understands the scene. As we can see in Figure 7.9, BLIP creates a short and safe

caption: “A group of people playing with a dog in a field.” This sentence is correct, but it does not mention some important things like the soccer ball or the goat. BLIP-2 gives a better description, talking about the soccer game and the goat, but it does not see the dog. GIT adds even more detail and says there is a goat and a soccer game, but it makes a small mistake by repeating “a goat and a goat.”

Our model, which uses a Vision Transformer with attention and an GRU decoder, gives the caption: “Two children are playing with a dog in a field.” This is a simple but correct sentence. It finds the children and the dog, but it misses the soccer ball and the goat. This shows that our model works well with the data it was trained on, but it cannot understand the whole scene as well as the larger models like BLIP-2 and GIT.

This naturally raises the next question: if models can already describe the same image in different ways, what happens when we start guiding them? What if we change not the image, but the prompt? This brings us to promptable vision-language models, which don’t just generate a caption, but generate a right caption for a right purpose.

7.2.1 Prompt-Based Captioning with LLaVA

With newer vision-language models, image captioning is no longer limited to a single neutral and fixed description of a scene. Models like LLaVA let us guide captioning with prompts, so the same image can produce very different captions depending on the instruction. This allows for a broader evaluation: not just whether a caption matches the references, but whether it meets the purpose set by the prompt.

To explore this, we designed eight prompts. Each represented more than a wording change, it reflected a different perspective and use of captioning:

1. **Baseline:** “Describe this image in one sentence.” Provides a simple, factual description that serves as our baseline, similar to Flickr8k-style captions.
2. **Detailed:** “Identify the main objects, their attributes, and the main action.” Encourages specificity while avoiding speculation.
3. **Relationships:** “Describe the relationships between objects and their spatial arrangement.” Emphasizes on spatial understanding and object interactions.
4. **Contextual:** “Explain what is happening in the scene and the likely context, but only based on what is visibly evident.” Combines context with what can actually be seen.
5. **Accessibility:** “Write a caption in the style of alternative text (alternative text) used by screen readers: concise, factual, and avoiding speculation.” Frames captioning as a tool for inclusive communication, highlighting its practical application in accessibility contexts.
6. **Comprehensive:** “Generate a longer caption that lists all visible objects, actions, and attributes in detail.” Tests whether the description is complete.
7. **Social-media:** “Write a catchy social media caption for this image.” Tests tone adaptation and audience awareness, another purpose specific prompt.
8. **Question:** “What is happening in this image?” Uses a question-answering style to focus on temporal and action understanding.

Taken together, these prompts cover a range of angles: from simple description to how objects relate in space, how actions are interpreted, and how captions can be adapted for different audiences.

Figure 7.10 shows how these prompts lead to very different outputs for the same image. More figure examples can be found in Appendix A.2.



Figure 7.10: Image used for prompt-based generation with LLaVA.

Reference: A black and white dog jumps to get the Frisbee.

Baseline: A black and white dog is playing with a frisbee.

Detailed: A black and white dog is jumping in the air to catch a yellow frisbee.

Relationships: A black and white dog is jumping in the air with a frisbee in its mouth.

Contextual: A black and white dog is jumping in the air to catch a yellow frisbee. The dog is in a backyard, and the frisbee is positioned in front of it. The scene likely represents a playful moment between the dog and its owner or a game of fetch.

Accessibility: A black and white dog is jumping in the air to catch a yellow frisbee.

Comprehensive: A black and white dog is jumping in the air to catch a yellow frisbee. The dog is wearing a collar and has a black tail. The frisbee is in the air above the dog, and the dog is in the middle of the jumping.

Social-media: “Frisbee dog in action! 🐾 🏃, ”

Question: A dog is playing with a frisbee.

Looking at Figure 7.10, we can observe how the same scene is described in very different ways depending on the prompt. The *Baseline* caption gives a simple factual statement, but it misses the sense of motion that defines the image. The *Detailed* caption adds the jumping action and the frisbee’s color, making it a fuller description of what is visible. The *Relationships* caption introduces a small mistake by saying the frisbee is already in the dog’s mouth, showing that describing positions can still be difficult. The *Contextual* caption goes further, describing the action and suggesting context, but it also risks adding hallucinations, such as the mention of the dog’s owner. The *Accessibility* caption stays concise and factual, balancing clarity and informativeness, which makes it suitable for alternative text. The *Comprehensive* caption expands on nearly every detail, including the collar, tail, and positioning of objects, which improves completeness but also leads to some repetition. The *Social-media* caption

shifts focus to style and tone, using emojis and playful language, but this makes it diverge from the reference captions typically used in evaluation. Finally, the *Question* prompt frames the caption as an answer, generating a short, concise description that emphasizes the action at the center of the scene.

To evaluate performance, we measured both traditional captioning metrics (BLEU, METEOR, ROUGE, CIDEr) and embedding-based metrics (CLIPScore, BLIPScore). Table 7.8 summarizes the results.

Table 7.8: Prompted captioning results using LLaVA on 1000 Flickr8k images

Prompt	BLEU-1	BLEU-4	METEOR	CIDEr	ROUGE-L	CLIPScore	BERTScore
Baseline	0.695	0.282	0.285	0.618	0.503	0.336	0.937
Detailed	0.654	0.256	0.285	0.603	0.489	0.331	0.929
Relationships	0.594	0.216	0.272	0.550	0.467	0.344	0.901
Contextual	0.323	0.095	0.241	0.147	0.337	0.364	0.791
Accessibility	0.671	0.265	0.282	0.577	0.493	0.339	0.892
Comprehensive	0.276	0.081	0.231	0.015	0.288	0.382	0.917
Social-media	0.368	0.076	0.179	0.170	0.257	0.218	0.814
Question	0.645	0.263	0.272	0.490	0.482	0.341	0.901

We observe the following: First, the *Baseline*, *Detailed*, and *Accessibility* prompts align most closely with the reference captions, achieving the highest BLEU, METEOR, and CIDEr scores. This is expected, since these instructions produce captions similar in style to the dataset. Second, prompts that demand more context (*Contextual*) or verbosity (*Comprehensive*) perform poorly under reference-based metrics, but achieve the highest CLIPScore, reflecting that they are semantically aligned with the image even if they deviate lexically. Finally, purpose-driven prompts like *Social-media* demonstrate LLaVA’s adaptability, though evaluation remains limited by the mismatch between prompt’s shift and traditional metrics.

Hence, we deduce that LLaVA is clearly able to follow different instructions and adjust its output. But our standard metrics don’t really reflect that flexibility. They’re good for measuring similarity to a fixed caption, but not for judging whether a caption is appropriate for a particular purpose.

To address this, we performed a final evaluation using an LLM as a judge. We chose GPT-5 mini [14], prompting it to act as an image-caption evaluator (prompt details can be found in Appendix A.3). The model assessed captions according to *Factual Accuracy*, *Relevance*, *Specificity*, *Conciseness*, and *Fluency*.

Table 7.9: Average LLM-as-a-judge evaluation across different prompt types (scores 1–5)

Prompt Type	GPT5-Score	Factual	Relevance	Specificity	Conciseness	Fluency
Baseline	3.29	3.28	3.76	2.80	4.87	4.87
Detailed	3.26	3.24	3.77	3.02	4.83	4.89
Relationships	3.18	3.21	3.79	2.84	4.78	4.88
Contextual	3.43	3.44	4.14	3.41	4.12	4.87
Accessibility	3.39	3.36	3.80	2.93	4.88	4.87
Comprehensive	3.50	3.49	4.30	3.78	4.17	4.82
Social-media	2.77	2.66	3.22	2.57	4.52	4.83
Question	3.24	3.26	3.64	2.49	4.91	4.88

Several observations stand out from this evaluation. The most notable is that the *Comprehensive* and *Contextual* prompts achieved the highest overall scores according to the LLM judge. Both produced captions that were longer and more descriptive, which naturally boosted their ratings in relevance and specificity. This is an interesting contrast to the reference-based metrics, where these same prompts performed poorly because their outputs diverged from the short captions in the dataset. In other words, when evaluated against human-like criteria rather than strict overlap with references, richer captions are rewarded rather than penalized.

Shorter prompts such as *Baseline*, *Accessibility*, and *Question* scored lower overall, but excelled in conciseness and fluency. This shows that brevity and clarity can be strong qualities, even if they leave out some visual details. The LLM scores reflect this trade-off: concise captions read well but may not always feel complete.

We also asked the LLM to assess whether our purpose-driven prompts (*Accessibility* and *Social-media*) actually fulfilled their intended goals. For this, we introduced two additional metrics: *Accessibility Fit* and *Social Media Fit*. Table 7.10 shows the results.

Table 7.10: Accessibility Fit and Social Media Fit scores across different prompt types (scores 1–5)

Prompt Type	Accessibility Fit	Social Media Fit
Accessibility	3.16	2.76
Baseline	3.06	2.82
Detailed	3.23	2.84
Contextual	3.38	3.11
Social-media	2.22	3.94
Question	2.86	2.63
Relationships	3.07	2.66
Comprehensive	3.68	3.10

When we looked specifically at purpose-driven criteria, an interesting observation appeared. The highest score for *Accessibility Fit* came not from the *Accessibility* prompt but from the *Comprehensive* prompt. While accessibility guidelines often tests conciseness, this finding suggests that overly short captions can exclude important cues. Richer descriptions, like those in *Comprehensive* outputs, may provide more context and therefore be more useful to people relying on alternative text. For *Social Media Fit*, the dedicated prompt performed best, as expected, as it encouraged a more appealing and stylistic output, including emojis and hashtags.

Overall, the results show that there is no single “best” prompting strategy. Instead, each prompt type is suited to a different goal: *Baseline* and *Detailed* give straightforward factual captions, *Comprehensive* and *Contextual* produce more detailed descriptions, *Accessibility* stays concise and clear, and *Social-media* produces captions with a more charming style. This shows why captions should be evaluated not only with reference-based metrics, but also with criteria that reflect their intended purpose and context.

Chapter 8

Conclusions and Future Work

This thesis explored image captioning through three main approaches: traditional encoder-decoder architectures, large pretrained vision-language models, and prompt-based captioning with instruction-following models. The experiments revealed both the strengths and limitations of each approach, as well as the challenges of evaluation.

8.1 Conclusions

For the classical models, results showed that architectural choices have a clear impact on caption quality. The Vision Transformer combined with additive attention and a GRU decoder consistently outperformed convolutional encoders such as ResNet and EfficientNet. Pretrained embeddings like GloVe helped recover rare words, while trainable embeddings adapted more effectively to the dataset. Beam search improved sentence quality compared to greedy decoding, but with added computational cost. These findings confirm that even on a relatively small dataset like Flickr8k, design decisions in the encoder, decoder, and training pipeline matter significantly.

The pretrained VLMs (BLIP, BLIP-2, and GIT) performed better overall than classical models in zero-shot settings, thanks to large-scale pretraining. They produced fluent captions and scored highly on semantic similarity measures. However, they also revealed issues such as hallucination and omission of important details. This underlines a trade-off: large models generalize well across varied content but can lose precision on specific visual details.

The prompt-based experiments with LLaVA demonstrated the flexibility of modern VLMs. By changing the prompt, the same model produced captions tailored for different intentions (factual descriptions, contextual descriptions, comprehensive listings, alt text for accessibility, stylistic captions for social media or question-answering). Standard reference-based metrics, however, favored captions closest to the dataset style, penalizing outputs that were longer, richer, or more creative. To address this, we introduced an additional evaluation with an LLM acting as a judge. This method captured qualities such as factual accuracy, relevance, clarity, fluency, and fitness for a specific purpose. The results showed that comprehensive and contextual captions, while scoring poorly against fixed references, were scored higher for informativeness and relevance. Conversely, short captions were praised for clarity and fluency but lacked detail. The purpose-specific metrics (Accessibility fit and Social-media fit) added further insight: accessibility was rated higher for comprehensive captions, since the additional detail provided more context, while the social-media prompt unsurprisingly performed best for social-media fit, thanks to its use of emojis and hashtags.

Overall, the experiments show that there isn't a single "best" way to generate image captions. Classical encoder-decoder models give stable and interpretable results, but they're limited by the size of the dataset. Pretrained vision-language models produce fluent and general captions, yet they can sometimes miss details. Prompt-driven models add a lot of flexibility, letting the same model generate captions for different purposes, but they also need new evaluation approaches beyond traditional reference-based metrics. Taken together, these findings suggest that image captioning should be seen as a purpose-driven task, where captions are judged not just on how similar they are to reference captions, but also on how well they serve their intended use.

8.2 Future Work

There are several directions worth exploring in future research:

- **Larger datasets:** Running experiments on larger datasets like MS COCO [38] or Conceptual Captions [57] to test how well the models scale and generalize.
- **Human evaluation:** While using an LLM as a judge gave useful insights, direct human

evaluation would provide stronger validation, especially for purpose-specific captions.

- **VQA-guided captioning:** Integrating visual question answering [58] could help generate captions with more targeted and detailed information.
- **Multilingual captioning:** Expanding caption generation to multiple languages could improve accessibility and broaden the practical use of these models.

Bibliography

- [1] Satish Kumar Satti, Goluguri N. V. Rajareddy, Prasad Maddula, and N. V. Vishnumurthy Ravipati. Image Caption Generation using ResNET-50 and LSTM. In *Proceedings of the 2023 IEEE Silchar Subsection Conference (SILCON)*, 2023.
- [2] Pooja Yadav, Neeraj Menon, Vinayakumar Ravi, Sowmya Vishvanathan, and Tuan D. Pham. A two-stage deep learning framework for image-based android malware detection and variant classification. *Comput. Intell.*, 38(5):1748–1771, 2022.
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [5] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.
- [6] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.
- [7] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation.

- In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022.
- [8] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A generative image-to-text transformer for vision and language. *Trans. Mach. Learn. Res.*, 2022, 2022.
- [9] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 2023.
- [10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL, 2014.
- [12] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [13] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021*

- Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7514–7528. Association for Computational Linguistics, 2021.
- [14] OpenAI. Gpt-5 system card. <https://cdn.openai.com/gpt-5-system-card.pdf>, August 2025.
- [15] Ali Farhadi, Seyyed Mohammad Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David A. Forsyth. Every picture tells a story: Generating sentences from images. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, volume 6314 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2010.
- [16] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1143–1151, 2011.
- [17] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164. IEEE Computer Society, 2015.
- [18] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):664–676, 2017.
- [19] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015.

- [20] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. Computer Vision Foundation / IEEE Computer Society, 2018.
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.
- [22] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280, 1989.
- [23] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society, 2017.
- [24] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society, 2015.
- [25] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.
- [26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

- [27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.
- [28] Ting-Yao Hsu, Chieh-Yang Huang, Ryan A. Rossi, Sungchul Kim, C. Lee Giles, and Ting-Hao Kenneth Huang. GPT-4 as an effective zero-shot evaluator for scientific figure captions. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5464–5474. Association for Computational Linguistics, 2023.
- [29] Xiujuan Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *CoRR*, abs/2004.06165, 2020.
- [30] Pengchuan Zhang, Xiujuan Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5579–5588. Computer Vision Foundation / IEEE, 2021.
- [31] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. *CoRR*, abs/2108.10904, 2021.
- [32] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 2021.

- [33] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *CoRR*, abs/2204.14198, 2022.
- [34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [35] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [36] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Nils Johan Bertil Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [37] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *Int. J. Comput. Vis.*, 123(1):74–93, 2017.

- [38] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.
- [39] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002.
- [40] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26, Barcelona, Spain, 2004. Association for Computational Linguistics.
- [41] Satanjeev Banerjee and Alon Lavie. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare R. Voss, editors, *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics, 2005.
- [42] David M. Chan, Suzanne Petryk, Joseph Gonzalez, Trevor Darrell, and John F. Canny. CLAIR: evaluating image captions with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13638–13646. Association for Computational Linguistics, 2023.
- [43] Mingqi Gao, Xinyu Hu, Jie Ruan, Xiao Pu, and Xiaojun Wan. LLM-based NLG evaluation: Current status and challenges. *CoRR*, abs/2402.01383, 2024.
- [44] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*, chapter 3.7, pages 107–111. O’Reilly Media, Inc., 2009.
- [45] François Chollet et al. Keras. <https://keras.io>, 2015.

- [46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.
- [47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [48] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019.
- [49] Poonam Bansal, Kiran Malik, Sharvan Kumar, and Charu Singh. EfficientNet-based image captioning system. In *2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT)*, pages 643–647, 2023.
- [50] Google Research and Hugging Face. Pretrained vision transformer model: google/vit-base-patch16-224-in21k. <https://huggingface.co/google/vit-base-patch16-224-in21k>, 2025.
- [51] David Graff and Christopher Cieri. English Gigaword. LDC2003T05, Web Download, 2003.
- [52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.
- [53] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P.

- Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* ChatGPT quality. <https://lmsys.org/blog/2023-03-30-vicuna/>, March 2023.
- [54] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.
- [55] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.
- [56] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [57] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565. Association for Computational Linguistics, 2018.
- [58] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. PromptCap: Prompt-guided task-aware image captioning. *CoRR*, abs/2211.09699, 2022.

APPENDICES

Appendix A

Additional Material

A.1 Additional Attention Visualizations

To complement the main example in Chapter 7, we include here further visualizations of additive attention during caption generation. The goal is to show both successful cases, where attention follows objects in a meaningful way, and failure cases, where the caption diverges from the scene even though the attention maps appear correct. In each figure, the original image is shown in (a), while the following subfigures (b), (c), ... illustrate the attention distribution for each predicted word.

Predicted: "a young girl does a handstand on the beach"

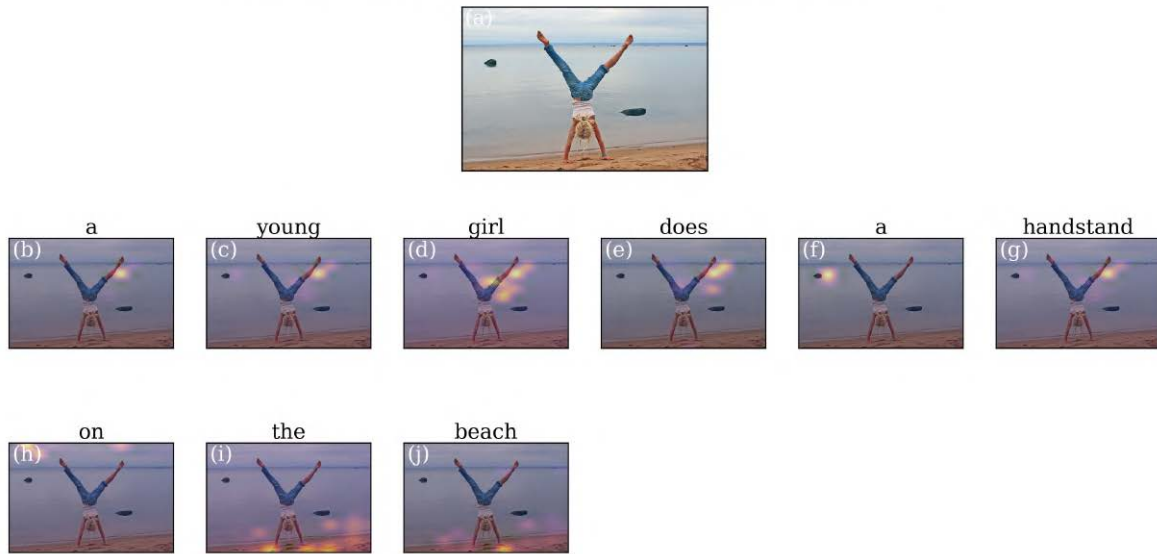


Figure A.1: Example 1: Predicted caption “a young girl does a handstand on the beach.”

The attention follows the girl when generating words like “girl” (d) and “handstand” (g), while it shifts to the sand when predicting “beach” (j). This shows how the model can localize different elements of the scene as the sentence unfolds.

Predicted: "a black dog is running on the beach next to the ocean"



Figure A.2: Example 2: Predicted caption “a black dog is running on the beach next to the ocean.”

The model correctly focuses on the dog for “dog” (d) and “running” (f), while for “beach” (i) and “ocean” (m) it shifts toward the background. Some steps, such as “on” (g) or “the” (h), produce more diffuse attention maps, reflecting the weaker grounding of function words.

Predicted: "a brown dog is jumping in the air to catch a brown dog"



Figure A.3: Example 3 (failure case): Predicted caption “a brown dog is jumping in the air to catch a brown dog.”

Here the model repeatedly attends to the dog throughout the sequence (e.g., “dog” in (d), “jumping” in (f), “catch” in (k)), which is visually correct. Although the attention looks at the right area, which includes both the dog and the girl, the model still predicts “dog” (n) again instead of “girl.” This can happen because words like dog appear more often in the dataset than girl, so the model prefers the frequent word. It can also be that the dog visually dominates the region, making it harder to detect the girl. Finally, sequence models often repeat strong words once they have been generated. This example shows that even when attention is correct, the caption can still be wrong.

A.2 Additional Prompt-Based Examples



Figure A.4: Image used for prompt-based generation with LLaVA.

Reference: A little girl climbing into a wooden play-house.

Baseline: A little girl climbs a ladder to a small door.

Detailed: A little girl is climbing a ladder to a small door.

Contextual: A little girl is climbing a ladder to a small wooden house. The house has a window and a door, and there is a bird cage nearby. The scene likely takes place in a backyard or a similar outdoor setting.

Accessibility: A little girl climbs a ladder to a small wooden house.

Comprehensive: A little girl is climbing a ladder to a small wooden house. Inside the house, there is a bird cage and a potted plant. The girl is wearing a pink dress and sandals.

Social-media: Little girl climbing a ladder to a play-house with a chicken coop in the background. #play-house #chickencoop #littlegirl

Question: A little girl is climbing a ladder to a play-house.

Relationships: A little girl is climbing a ladder to a small wooden house with a window and a bird cage.

Figure A.4 shows how different prompts can guide the model to produce captions of varying style and detail for the same image. The baseline remains short and generic, while prompts such as *Contextual* or *Comprehensive* push the model to add relevant details about the house, bird cage, and setting. Prompts aimed at specific purposes, such as *Accessibility* or *Social-media*, adapt the caption toward clarity or informality, respectively. This example shows how prompt-based captioning can successfully tailor the description to different use-cases while staying faithful to the scene.



Figure A.5: Image used for prompt-based generation with LLaVA.

Reference: A man and woman pose for the camera while another man looks on.

Baseline: A man and a woman are posing for a picture.

Detailed: A man and a woman are standing together, the man is holding a piece of food in his mouth.

Contextual: A man and a woman are standing next to each other, with the man holding a piece of food in his mouth. They are both smiling, and the woman is holding the man's hand. The context of the scene could be a casual social gathering or a celebration, where the man is enjoying a snack while interacting with the woman.

Accessibility: A man and a woman are standing together and smiling.

Comprehensive: A man and a woman are standing together, with the man holding a piece of food in his mouth. The woman is smiling and has blonde hair. They are both wearing black and white clothing. The man is wearing a blue shirt and the woman is wearing a black shirt. They are standing in front of a building.

Social-media: Smiling faces and a shared snack moment!

Question: A man is eating a hot dog.

Relationships: A man and a woman are standing next to each other, with the woman on the right and the man on the left.

Figure A.5 shows a failure case where the prompts push the model in the wrong direction. Several outputs mention food, with the *Question* prompt even saying “a man is eating a hot dog.” In reality, the man is holding the woman's hand, and there is no food at all. The baseline and accessibility captions are closer to the scene but still fail to capture the gesture. This example illustrates how prompts can also mislead the model and generate inaccurate descriptions.

A.3 LLM as a Judge Evaluation Criteria Prompt

You are an objective judge for image captions. Use the following criteria:

General scoring (1-5):

- 5 - Excellent: fully accurate, specific, complete, concise, and fluent; no unsupported inference.
- 4 - Good: accurate and fluent, minor missing detail or small inference.
- 3 - Adequate: mostly correct but missing important visible detail or contains a small incorrect/unsupported element.
- 2 - Poor: contains multiple inaccuracies or significant unsupported inferences.
- 1 - Bad: irrelevant, misleading, or contradicts the image.

General evaluation criteria:

- Factual accuracy
- Relevance/coverage
- Specificity
- Conciseness/clarity
- Fluency/grammar
- Hallucination flag

Extra purpose-fit criteria (only if applicable to the caption type):

- `accessibility_fit` (1-5): How well this caption works as alt-text for a blind/low-vision user. Must be clear, descriptive, and avoid fluff.

- `social_media_fit` (1-5): How well this caption works as a natural social media post. Must be engaging, catchy, may use emojis/hashtags.

Output JSON format exactly like this:

```
"score": int,  
"factual_accuracy": int,  
"relevance": int,  
"specificity": int,  
"conciseness": int,  
"fluency": int,  
"hallucination_flag": bool,  
"accessibility_fit": int,  
"social_media_fit": int
```

Only output valid JSON, nothing else.