

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Viktar ATLIHA

IMPROVING IMAGE CAPTIONING METHODS USING MACHINE LEARNING APPROACHES

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
INFORMATICS ENGINEERING (T 007)

Vilnius, 2023

The doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2018–2023.

Scientific Supervisor

Prof. Dr Dmitrij ŠEŠOK (Vilnius Gediminas Technical University, Informatics Engineering – T 007).

The Dissertation Defense Council of the Scientific Field of Informatics Engineering of Vilnius Gediminas Technical University:

Chairperson

Prof. Dr Dalius MAŽEIKI (Vilnius Gediminas Technical University, Informatics Engineering – T 007).

Members:

Assoc. Prof. Dr Nikolaj GORANIN (Vilnius Gediminas Technical University, Informatics Engineering – T 007),

Prof. Dr Rytis MASKELIŪNAS (Kaunas University of Technology, Informatics Engineering – T 007),

Dr Viktor MEDVEDEV (Vilnius University, Informatics Engineering – T 007),

Prof. Dr Audris MOCKUS (University of Tennessee, USA, Informatics – N 009).

The dissertation will be defended at the public meeting of the Dissertation Defense Council of the Scientific Field of Informatics Engineering in the Senate Hall of Vilnius Gediminas Technical University at **2 p. m. on 12 June 2023**.

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.

Tel.: +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vilniustech.lt

A notification on the intended defense of the dissertation was sent on 11 May 2023. A copy of the doctoral dissertation is available for review at the Vilnius Gediminas Technical University repository <http://dspace.vgtu.lt>, at the Library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania) and the Library of Kaunas University of Technology (K. Donelaičio g. 20, LT-44239 Kaunas, Lithuania).

Vilnius Gediminas Technical University book No 2023-021-M

doi:10.20334/2023-021-M

©Vilnius Gediminas Technical University, 2023

©Viktar Atliha, 2023

viktar.atliha@vilniustech.lt

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Viktar ATLIHA

VAIZDŲ ANTRAŠČIŲ GENERAVIMO METODŲ TOBULINIMAS MAŠININIO MOKYMOSI METODAIS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
INFORMATIKOS INŽINERIJA (T 007)

Vilnius, 2023

Disertacija rengta 2018–2023 metais Vilniaus Gedimino technikos universitete.

Vadovas

prof. dr. Dmitrij ŠEŠOK (Vilniaus Gedimino technikos universitetas,
informatikos inžinerija – T 007).

Vilniaus Gedimino technikos universiteto Informatikos inžinerijos mokslo krypties
disertacijos gynimo taryba:

Pirmininkas

prof. dr. Dalius MAŽEIKAS (Vilniaus Gedimino technikos universitetas,
informatikos inžinerija – T 007).

Nariai:

doc. dr. Nikolaj GORANIN (Vilniaus Gedimino technikos universitetas,
informatikos inžinerija – T 007),

prof. dr. Rytis MASKELIŪNAS (Kauno technologijos universitetas,
informatikos inžinerija – T 007),

dr. Viktor MEDVEDEV (Vilniaus universitetas, informatikos inžinerija –
T 007),

prof. dr. Audris MOCKUS (Tenesio universitetas, JAV, informatika – N 009).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertaci-
jos gynimo tarybos posėdyje **2023 m. birželio 12 d. 14 val.** Vilniaus Gedimino
technikos universiteto senato posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva.

Tel.: (8 5) 274 4956; faksas (8 5) 270 0112; el. paštas doktor@vilniustech.lt

Pranešimai apie numatomą ginti disertaciją išsiųsti 2023 m. gegužės 11 d. Diser-
taciją galima peržiūrėti Vilniaus Gedimino technikos universiteto talpykloje
<http://dspace.vgtu.lt>, Vilniaus Gedimino technikos universiteto bibliotekoje
(Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) ir Kauno technologijos universiteto
bibliotekoje (K. Donelaičio g. 20, LT44239 Kaunas, Lietuva).

Abstract

Recently, computer vision (CV) and natural language processing (NLP) fields started gaining increasing attention from researchers and the industry. While the first bunch of methods allows for solving many tasks within the images and pictures domain, such as image classification, image detection, etc., the others work in a text domain, including text classification or translation tasks. However, many problems remain on a border between the two mentioned domains that have a practical use. One of them is called image captioning. The goal of image captioning systems is to automatically generate a human-like textual description of the given image. Such systems could be used for smoother human–computer interactions, information retrieval, or, more importantly, to help visually impaired people. To succeed, algorithms used in these systems should consume low resources (particularly, acquire little memory) and be of high quality.

As the image captioning task is a cross-domain, and state-of-the-art models for computer vision and natural language processing tasks use deep learning models, it also leads to using such approaches for the image captioning task. However, most of the well-known methods of improving image captioning models tend to be focused more on quality improvement, considering no additional resources are needed. Thus, the best models, for now, are very big and unsuitable for use on mobile and other memory-constrained devices where they could bring the greatest practical benefit.

The dissertation consists of an introduction, three main chapters, and general conclusions. The First Chapter reviews existing research on image captioning. The Second Chapter investigates the application of model compression methods for existing image captioning models, proposing several methods of reducing the model size without significant quality loss. The Third Chapter focuses on improving image captioning models without significant changes (or without changes at all) in model architecture, highlighting the importance of such methods.

The performed experiments and analysis showed that image-captioning models could be significantly compressed without almost any quality loss. Application of all proposed methods allowed to reduce the model size by 91%, losing only up to 3% in the main quality metrics. More than that, methods proposed for improving quality without changing models' architecture allowed for almost neutralizing this effect, leading to up to 5% quality improvements.

Reziumė

Pastaruoju metu kompiuterinio regėjimo ir natūralios kalbos apdorojimo sritys su-
laukia vis daugiau mokslininkų ir pramonės dėmesio. Pirmos srities metodų grupė
leidžia išspręsti daugybę užduočių, susijusių su vaizdų apdorojimu, pavyzdžiui,
vaizdų klasifikavimą, vaizdų aptikimą ir kt., antros srities metodai apdoroja tekstą,
įskaitant teksto klasifikavimo ar vertimo užduotis. Tačiau yra ir uždavinių, esančių
šių dviejų sričių sankirtoje. Vienas iš tokių uždavinių yra vaizdų antraščių generavi-
mas. Vaizdų antraščių generavimo sistemos tikslas automatiškai sugeneruoti panašų
amogaus sudarytą tekstinį pateikto vaizdo aprašymą. Tokios sistemos galėtų būti
naudojamos sklandesnei žmogaus ir kompiuterio sąveikai, informacijos paieškai
arba, kas dar svarbiau, padėti silpnaregiams. Kad šiose sistemose naudojami algo-
ritmai būtų sėkmingi, jie turi sunaudoti mažai išteklių (ypač, turėti mažai atminties)
ir būti kokybiški.

Kadangi vaizdų antraščių generavimo užduotis yra kelių sričių sankirtoje, o
moderniausi modeliai, skirti tiek kompiuterinio regėjimo, tiek natūralios kalbos ap-
dorojimo užduotims, naudoja giliojo mokymosi modelius, tokius metodus reikia nau-
doti ir vaizdų antraštėms generuoti. Tačiau dauguma gerai žinomų vaizdų antraščių
generavimo modelių tobulinimo būdų yra labiau orientuoti į kokybės gerinimą,
neatsižvelgiant į jokių papildomus reikalingus išteklius. Taigi geriausi modeliai
šiuo metu yra labai dideli ir netinkami naudoti mobiliuosiuose ir kituose ribotos
atminties įrenginiuose, kur galėtų duoti didžiausią praktinę naudą.

Disertaciją sudaro įvadas, trys pagrindiniai skyriai ir bendrosios išvados. Pir-
majame skyriuje apžvelgiami esami vaizdų antraščių generavimo metodų tyrimai.
Antrajame skyriuje nagrinėjamas modelių glaudinimo metodų taikymas esamiems
vaizdų antraščių generavimo modeliams, siūlomi keli modelio dydžio sumaži-
nimo būdai kokybės neprarandant. Trečiajame skyriuje dėmesys telkiamas į vaizdų
antraščių generavimo modelių tobulinimą be reikšmingų modelių architektūros
pokyčių, pabrėžiant tokių metodų svarbą.

Atlikti eksperimentai ir analizė parodė, kad vaizdų antraščių generavimo mode-
lius galima labai suspausti, beveik neprarandant kokybės. Visų siūlomų metodų
taikymas leido sumažinti modelio dydį 91 %, prarandant tik iki 3 % pagrindinėje
kokybės metrikroje. Be to, pasiūlyti metodai pagerinti kokybę, nekeičiant modelių
architektūros, leido beveik neutralizuoti šį poveikį, todėl kokybė pagerėjo iki 5 %.

Notations

Abbreviations

BERT – Bidirectional Encoder Representations from Transformers.
BLEU – Bilingual Evaluation Understudy.
CIDEr – Consensus-Based Image Description Evaluation.
CLIP – Contrastive Language–Image Pre-training.
CNM – Collocate Neural Model.
CV – Computer Vision.
DLCT – Dual-Level Collaborative Transformer.
ETA – EnTangled Attention.
GCN – Graph Convolutional Network.
GloVe – Global Vectors for Word Representation.
GSA – Geometry-aware Self-Attention.
HMM – Hidden Markov Model.
HP – Hierarchy Parsing.
LSTM – Long Short-Term Memory.
METEOR – Metric for Evaluation of Translation with Explicit Ordering.
MS COCO – Microsoft Common Objects in Context.

NLP – Natural Language Processing.

R-CNN – Region-Based Convolutional Neural Network.

RNN – Recurrent Neural Network.

ROUGE-L – Recall-Oriented Understudy for Gisting Evaluation.

SCA-CNN – Spatial and Channel-Wise Attention in Convolutional Networks.

SGAE – Scene Graph Auto-Encoder.

SPICE – Semantic Propositional Image Caption Evaluation.

VDR – Visual Dependency Representation.

Word2vec – Word to Vector.

Contents

INTRODUCTION	1
Problem Formulation	1
Relevance of the Dissertation	2
The Object of the Research	2
The Aim of the Dissertation	2
The Objectives of the Dissertation	2
Research Methodology	3
The Scientific Novelty of the Dissertation	3
The Practical Value of the Research Findings	3
The Defended Statements	4
Approval of the Research Findings	4
The Structure of the Dissertation	5
1. LITERATURE REVIEW ON IMAGE CAPTIONING METHODS	7
1.1. Image Captioning Models	7
1.1.1. Template-based Methods	8
1.1.2. Retrieval-based Methods	9
1.1.3. Encoder-Decoder Architecture	10
1.1.4. Visual Models	12
1.1.5. Language Models	19
1.1.6. Joint Visual-Language Models	23

1.1.7. Generative Adversarial Networks for Image Captioning	25
1.2. Image Captioning Model Training and Evaluation Methods	26
1.2.1. Image Captioning Datasets	28
1.2.2. Image Captioning Metrics	29
1.2.3. Image Captioning Task Variations	34
1.3. Image Captioning Practical Problems	36
1.4. Neural Network Model Compression Techniques	37
1.4.1. Efficient Architectures	37
1.4.2. Neural Network Pruning Methods	38
1.4.3. Neural Network Quantization Methods	39
1.4.4. Image Captioning Model Knowledge Distillation	41
1.4.5. Image Captioning Model Compression Methods	42
1.5. Augmentation Techniques for Dataset Extension	43
1.6. Conclusions of the First Chapter and Formulation of the Dissertation Tasks	44
2. REDUCING IMAGE CAPTIONING NEURAL NETWORK MODELS' MEMORY CONSUMPTION	45
2.1. Experiment Setup	46
2.2. Encoder Compression	46
2.2.1. Encoders for Image Captioning	47
2.2.2. Empirical Evaluation and Results and Discussion of Empirical Evaluation of Encoder Choice	47
2.3. Decoder Compression	47
2.3.1. Architecture Hyperparameter Changes for Compression	48
2.3.2. Empirical Evaluation and Discussion of Architectural Hyperparameter Choice Results	50
2.3.3. Decoder Neural Network Pruning	50
2.3.4. Empirical Evaluation Results and Discussion for Decoder Pruning	51
2.3.5. Decoder Neural Network Quantization	54
2.3.6. Empirical Evaluation Results and Discussion for Decoder Quantization	55
2.4. Image Captioning Model Knowledge Distillation	55
2.4.1. Empirical Evaluation Results and Discussion of Knowledge Distillation	58
2.5. Conclusions of the Second Chapter	63
3. INCREASING IMAGE CAPTIONING NEURAL NETWORK MODELS' QUALITY	65
3.1. Experiment Setup	66
3.2. Encoder Choice for Image Captioning Models	67

3.3. Pre-Trained Word Embeddings	70
3.3.1. One-hot Encoding	71
3.3.2. Word-to-Vector Embeddings	71
3.3.3. Global Vectors for Word Representation	72
3.3.4. Empirical Evaluation and Discussion of Results of the Word Embeddings Choice for Image Captioning Models.....	73
3.4. Captions Augmentation Methods for Dataset Extension	77
3.4.1. Synonymous Augmentation for Dataset Extension	77
3.4.2. Contextualized Word Embedding Augmentation for Dataset Extension	78
3.4.3. Empirical Evaluation of Augmentation Techniques	80
3.4.4. Results and Discussion of Augmentation Techniques	82
3.4.5. Quantitative Analysis of Augmentation Techniques	87
3.5. Caption Augmentations for Compressed Models	88
3.5.1. Augmentation with Knowledge Distillation	89
3.5.2. Empirical Evaluation of Augmentation with Knowledge Distil- lation	91
3.5.3. Results and Discussion of Augmentation with Knowledge Dis- tillation	91
3.6. Conclusions of the Third Chapter	93
GENERAL CONCLUSIONS	95
REFERENCES	97
LIST OF SCIENTIFIC PUBLICATIONS BY THE AUTHOR ON THE TOPIC OF THE DISSERTATION	111
SUMMARY IN LITHUANIAN	113

Introduction

Problem Formulation

Image captioning is the process of automatically generating a textual description of an image. The goal of this task is to create descriptions that are as similar as possible to how a human would describe the image. In recent years, most approaches to solving the image captioning task have involved the use of neural networks, with most of these architectures being of the encoder-decoder type.

The recent research focus in this field has primarily been on modifying the neural network architectures used for image captioning. However, as these models continue to improve in quality, their size also increases, making it difficult to use state-of-the-art models for image captioning on resource (especially memory) limited devices, such as mobile devices or wearable technologies, where there is a potential for a wide range of applications. Therefore, it is important to continue improving the quality of image captioning systems without increasing their size, as well as reducing their size without sacrificing a significant amount of quality, to make these systems more practical for real-world applications.

The dissertation is primarily concentrated on achieving the latter goal, that is, improving the quality of image captioning systems without increasing their size, as well as reducing their size without sacrificing a significant amount of quality.

Relevance of the Dissertation

Image captioning is a crucial task in the field of computer vision and has many practical applications, such as assisting visually impaired individuals, aiding in image search, and improving image understanding for artificial intelligence systems.

Reducing memory consumption without loss of quality is a critical aspect of image captioning model development, especially on memory-limited devices such as smartphones and embedded systems. Addressing this challenge can enable the deployment of image captioning models on memory-limited devices, making them more accessible to a wider range of users.

Similarly, improving the quality of image captioning models without significant changes in architecture is also a crucial research area. While complex architectures and techniques can improve the accuracy of models, they often come with increased computational costs and complexity. Therefore, finding ways to improve the quality of image captioning models while maintaining a simple and efficient architecture can significantly impact the practicality and usefulness of these models.

The Object of the Research

The object of the study is dataset augmentation methods for improving the quality of image captioning models' and image captioning models' compression methods.

The Aim of the Dissertation

The study aims to improve the quality metrics of image captioning systems and decrease image captioning model size without significant loss of quality.

The Objectives of the Dissertation

To achieve the dissertation's aim, the following tasks must be solved:

1. To propose and evaluate model compression methods to reduce image captioning model size by more than 80% without a significant quality loss of more than 5%.
2. To create a text augmentation method for extending existing datasets and improving the quality of existing image captioning systems.

Research Methodology

This dissertation used the quantitative methodology. A literature review of current methods has been conducted to get a better insight into the possibilities for improvement of the image captioning systems. Several improvements to the existing models have been proposed, as well as new methods of model compression and quality improvements. Then, an evaluation of the proposed methods was conducted to determine the best-performing machine learning and deep learning methods. Finally, a metrics analysis was conducted to make conclusions on the effect of the proposed methods.

The Scientific Novelty of the Dissertation

The scientific novelty of this study is specified as follows:

- The proposed use of a particular combination of deep learning model compression techniques, such as hyperparameters change, pruning, quantization, and knowledge distillation, for image captioning models.
- The augmentations of image captioning datasets using synonymous replacements and contextualized word embeddings proposed in the application to image captioning task.
- The proposed use of pre-trained Word2vec and GloVe pre-trained embeddings for the image captioning model.
- The text augmentation techniques combined with compression methods to achieve even better performance.

The Practical Value of the Research Findings

The findings of this research have both theoretical and practical significance. In the area of image captioning model compression, the results obtained in this study establish a strong baseline for further research and demonstrate that current state-of-the-art models are not using their size as effectively as they could be and that improvements can be made without increasing the size of these models. From a practical perspective, the best models obtained in this study are small enough to be used or potentially used on memory-restricted devices, which was not previously possible. By compressing image captioning models, it becomes possible to bring the benefits of this technology to a wider audience, including people who may not have access to high-end computing resources.

Additionally, model compression techniques can help reduce the environmental impact of deep learning models, which require significant amounts of energy to train and run. By making these models more efficient, researchers can help reduce the carbon footprint of the AI industry.

The introduction of the idea of using text augmentation for the image captioning task opens the possibility for researchers to investigate other augmentation methods and their impact on model quality. Several recent works that reference this research have already been published, indicating a high level of interest in this topic.

Finally, the evaluation of the image decoder's impact and pre-trained word embeddings on model quality shows that improvements can be made by modifying only small parts of the model without the need for complex new architectures or training methods. Overall, this research has practical value in terms of advancing the state-of-the-art in image captioning and making these models more accessible for real-world applications.

The Defended Statements

The following statements, based on the results of the present dissertation, may serve as the official hypotheses to be defended:

1. The use of deep learning models compression techniques (such as pruning, quantization, and knowledge distillation) for image captioning models in a combination proposed in the dissertation allows for reducing the model size by up to 91%, losing up to 3% in quality metrics.
2. The augmentations of image captioning datasets using synonymous replacements and contextualized word embeddings introduced particularly for use in image captioning systems increase quality by up to 5%, depending on the model.
3. The use of better decoders and pre-trained word embeddings, proposed in the dissertation, can affect image captioning methods' quality increase by up to 5%.

Approval of the Research Findings

The results of the dissertation were published in six scientific publications. Two were published in referred journals of *Clarivate Analytics* (also referred to as *Thomson Reuters*) *Web of Science* databases. Four were published in conference proceedings. The author also gave four presentations at scientific conferences in Lithuania and abroad:

- 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2020, Vilnius, Lithuania;
- 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, Vilnius, Lithuania;
- VII International scientific conference “High Technologies. Business. Society,” 2022, Borovets, Bulgaria;
- 2022 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2022, Vilnius, Lithuania.

The Structure of the Dissertation

The dissertation consists of an introduction, three main chapters, general conclusions, references and a list of the author’s publications.

The first chapter is about creating image captioning systems. It covers different approaches, typical architectures, joint visual-language methods, training methods, datasets, and evaluation metrics. It also discusses variations, neural network compression, and augmentation techniques

The second chapter focuses on reducing the size of image captioning neural network models without losing quality.

The third chapter of this dissertation is about enhancing the image captioning models’ performance without making substantial modifications to their architectures. The approach involves several techniques, such as modifying the encoder, utilizing pre-trained word embeddings, and applying data augmentation methods to the training dataset.

The total scope of the dissertation is 123 pages. There are 56 equations, 34 pictures, and 28 tables in the text. The dissertation text uses 169 references.

Literature Review on Image Captioning Methods

This chapter reviews the methods for creating image captioning systems. It begins with a brief description of the approaches to the image captioning task before neural networks were commonly used for this purpose. The chapter then introduces the typical image captioning architecture, based on the encoder-decoder framework. The key differences between various architectures that have been researched are described, first from the perspective of the encoder (visual model) and then from the perspective of the decoder (language model). The chapter then covers joint visual-language methods, which are different from the approaches previously described. The chapter continues by discussing training methods for image captioning models, followed by a sub-chapter on datasets and evaluation metrics. The chapter follows with an overview of some variations on the basic image captioning task. The chapter contains an overview of typical neural network compression methods and augmentation techniques for model quality improvements. The First Chapter ends with a summary of the main literature review findings.

1.1. Image Captioning Models

The image captioning task is at the intersection of computer vision (CV) and natural language processing (NLP). The goal of classical image captioning systems is to generate human-like textual descriptions based on visual input. As it combines both the visual and textual domains, image captioning is more complex than typical

in-domain tasks in either CV (such as classification and detection) or NLP (such as sentiment classification and translation). This means that high-quality image captioning systems should consist of both high-quality visual and textual subsystems and, that such systems can benefit from research in both domains. In fact, the image captioning task can be seen as a kind of language translation task, where the input “language” is the visual content of an image.

The task of generating sentences and phrases describing a photo or image has attracted a lot of research interest, especially after the first successful neural network approach was proposed by Vinyals et al. (2015). This is confirmed by many literature reviews published since 2015, such as Staniūtė and Šešok (2019), Stefanini et al. (2022), Hossain et al. (2019), Bernardi et al. (2016), Bai and An (2018), Liu et al. (2019b), Sharma et al. (2020), Kumar and Goel (2017), Shabir and Arafat (2018), and Jenisha and Purushotham (2016). Many of these reviews are from recent years, indicating the continued interest in this topic in the scientific community.

Before 2015, there were also some papers addressing the image captioning task. These can generally be divided into two main categories: template-based methods and retrieval-based methods.

1.1.1. Template-based Methods

The common idea behind template-based methods for image captioning is to generate a template consisting of more general concepts (such as parts of speech or broad categories) and then fill it in with more precise information. This two-step approach makes the task simpler because each step focuses on a more specific, less general problem. However, there is a wide variety of possible templates and ways to generate them.

For example, Kulkarni et al. (2011) proposed to construct a conditional random field based on the calculated image potentials, calculate the labeling of such a graph, and then generate a caption based on the labeling. Elliott and Keller (2013) introduced visual dependency representations (VDRs) and visual dependency grammar, and compared two methods: VDR-based and bag-of-region-based. It turns out that the VDR-based method allows for higher results. Kuznetsova et al. (2012) proposed using image-level content planning as the abstract generation. They detect objects in the image, then select which of them will be used for the caption, and order the selected objects in the same way as they will be used in the final caption. Yang et al. (2011) presented template entities as parts of speech, such as nouns, verbs, and prepositions, and then hidden Markov models (HMMs) were used to generate sentences. Kuznetsova et al. (2014), presented the template using a dependencies tree.

Mitchell et al. (2012) used templates not only for sentence construction but also for object detection. Thus, every image can be characterized by a triple $\langle A,$

$B, C >$ where A is a set of objects, B is a set of actions, and C is a set of relations. Similarly, every description can be viewed as $< A, B, C >$ where A corresponds to nouns, B corresponds to verbs, and C corresponds to prepositions. The method itself translates one triple into another.

Some of the approaches use more specific, image-related information found on the web but still base the caption generation on templates. Aker and Gaizauskas (2010) incorporated geotags relevant to the image and web documents containing information about the image's geolocation. Yao et al. (2010) used a web ontology to gather more information related to the image. They also used and-or graphs (AoGs) for the image captioning task and vocabularies of visual elements based on the image parsing engine, including parts of objects, scenes, primitives, and relations between objects, to construct the AoG. Once constructed, the semantic web ontology is used to generate the descriptions. Li et al. (2011) described the use of web-scale n -grams for the task. The approach consists of two stages: phrase selection and phrase fusion, both of which are based on n -grams.

Unlike the other methods in this subsection, Gupta et al. (2012) utilized semantic information contained in the dataset of image descriptions. The approach focuses on semantically and linguistically motivated phrases.

1.1.2. Retrieval-based Methods

Retrieval-based methods for image captioning do not generate descriptions from scratch, even in template form. Instead, as proposed in one of the first papers on retrieval-based approaches for image captioning by Pan et al. (2004), the image captioning task can be formulated as follows: "Given a set of images, where each image is captioned with a set of terms describing the image content, find the association between the image features and the terms." In other words, the task is not to generate descriptions but to find the appropriate caption from a predefined set of sentences.

To solve this task, several different methods have been proposed in different papers. For example, Pan et al. (2004) proposed converting the image into a number of "blob-tokens" (image regions) and constructing a matrix where the potential similarity between each pair of a "blob-token" and a word is estimated. This matrix is then used to find the optimal caption for the image by selecting the words with the highest scores.

Farhadi et al. (2010) described how to translate both images and sentences into a common space of "meanings" and find their similarity there. The authors of this paper propose using a neural network architecture called a "storytelling" model, which is trained to encode images and sentences into a shared semantic space and then use the resulting embeddings to find the most similar captions for a given image.

Similarly, to find corresponding captions, Ordonez et al. (2011) proposed using a global image representation for that purpose. The authors of this paper proposed using a pre-trained deep convolutional neural network to extract a compact global representation of an image, which is then used to retrieve the most similar caption from a predefined set of captions.

Frome et al. (2013) took this approach a step further by pre-training separate state-of-the-art deep learning models for both embedding images and words into similar spaces. The authors of this paper proposed using a novel “devise” model, which is trained to learn separate image and word embeddings that can be used to find the most similar captions for a given image.

Other methods for calculating such embeddings for caption retrieval have been researched in papers by Kiros et al. (2014a), Kiros et al. (2014b), Karpathy et al. (2014), Gong et al. (2014), and Sun et al. (2015). These papers propose various techniques for learning image and word embeddings that can be used to find the most similar captions for a given image, including methods based on neural networks, matrix factorization, and unsupervised learning.

1.1.3. Encoder-Decoder Architecture

The encoder-decoder framework is the most popular approach for end-to-end image captioning systems. This framework was popularized by one of the most highly-cited works in the image captioning field by Vinyals et al. (2015), and is illustrated in Fig. 1.1.

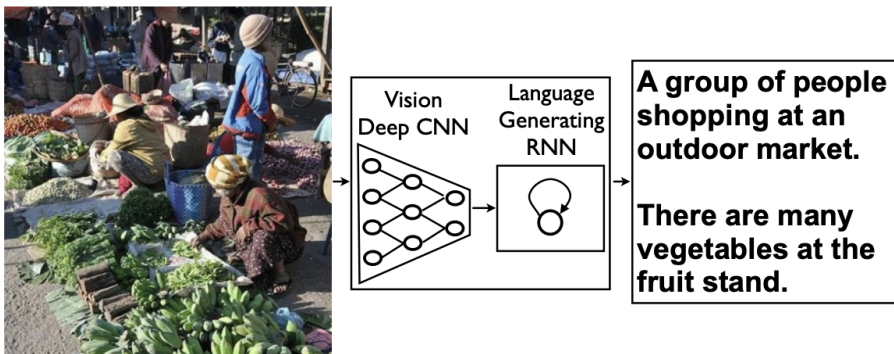


Fig. 1.1. Encoder-decoder architecture for image captioning introduced by Vinyals et al. (2015)

The typical encoder-decoder system consists of two parts: a visual understanding model, which acts as an encoder, and a language model, which acts as a decoder. The visual model is typically implemented as a large convolutional neural network, which can compress an image into a low-dimensional representation. The language model is typically implemented as a recurrent neural network, which can generate a sequence of tokens based on the image representation. Specific visual models used in image captioning are discussed in Sub-chapter 1.1.4. Language models commonly used for the image captioning task are described in Sub-chapter 1.1.5.

The goal of the encoder-decoder framework is to maximize the probability of generating a correct description of an image. This is typically achieved by optimizing the model parameters using the following objective function:

$$\theta^* = \arg \max_{(I,S)} \sum \log p(S|I; \theta), \quad (1.1)$$

where θ are the model parameters, I is an image, and S is a sentence. By using an encoder E to calculate a hidden image representation, and a decoder D to generate the description word-by-word (bounded by a maximum of N tokens), the probability $p(S|I)$ can be reformulated as follows:

$$\log p(S|I) = \sum_{t=0}^N p(S_t|E(I), S_0, S_1, \dots, S_{t-1}). \quad (1.2)$$

During the training phase, pairs (I, S) are ground-truth image-caption pairs from the dataset. These pairs are used to maximize the log-likelihood objective function defined above. However, different training methods have been proposed in the literature, which will be described in detail in Sub-chapter 1.2.

The inference phase of the encoder-decoder framework also follows a common pattern, as described by Vinyals et al. (2015). The generation process uses a sampling approach, where tokens are sampled sequentially based on the previously generated words and their corresponding embeddings, using the probability distribution $p(S|I; \theta)$. Sampling continues until the maximum sentence length is reached, or a special end-of-sentence token is sampled. This sequential generation method does not strictly generate the sentence with the maximum probability but is used to reduce computational complexity. To improve the quality of the generated captions, beam search is often used.

Beam search is a method for improving the quality of the generated captions by considering multiple potential captions and selecting the most likely ones at each time step. More formally, let $s_{i,j}$ represent the j -th most likely caption generated by the model at time step i . The probability of each caption is calculated using the $p(S|I; \theta)$ probability distribution. The beam search algorithm then selects the k

captions with the highest probabilities at each time step and generates the k most likely next words for each of these captions. The resulting set of k^2 captions of length $i + 1$ is then reduced to the k most probable captions. This process is repeated until the maximum sentence length is reached or a special end-of-sentence token is generated.

Overall, the encoder-decoder framework is a powerful approach for image captioning, as it allows for end-to-end training of the visual and language models and provides a flexible and modular framework for improving the quality of generated captions.

1.1.4. Visual Models

One of the most common visual models for image captioning is a large convolutional neural network (CNN) which is used as a decoder transforming image into a fixed-size embedding in a lower-dimensional space:

$$a = E(I), \quad (1.3)$$

where E is a CNN.

Architectures of such CNNs followed the progress in solving other computer vision tasks, such as image classification. So, the most frequently used architectures for image captioning are the same as those used for image classification: AlexNet by Krizhevsky et al. (2017), VGG by Simonyan and Zisserman (2015), ResNet by He et al. (2016), GoogleNet by Szegedy et al. (2015) and Inception-v3 by Szegedy et al. (2016) among which VGG and ResNet are the most popular.

AlexNet is one of the first CNN architectures that was used for the image classification task. It is a relatively shallow architecture, consisting of only eight convolutional, max pooling, and dense layers. The architecture of the model is illustrated in Fig. 1.2.

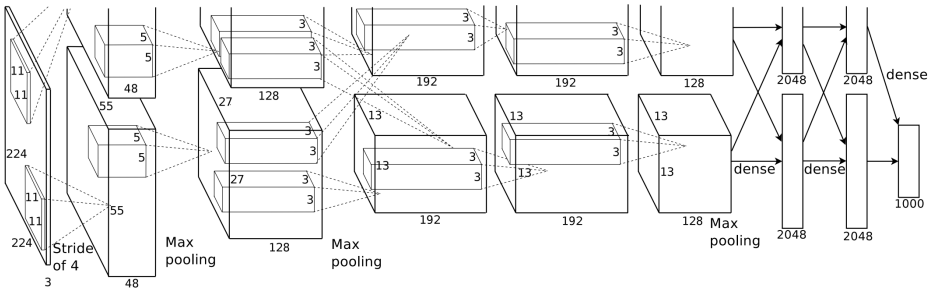


Fig. 1.2. AlexNet neural network architecture by Krizhevsky et al. (2017)

A later successor of AlexNet, based on the same ideas, is the VGG architecture, illustrated in Fig. 1.3. VGG is a classical, very deep architecture made up of 3×3 convolution filters, as well as pooling and fully connected layers. The VGG depth allows for the improvement of metrics and better internal image representations but comes at the cost of decreased inference speed and increased memory consumption.

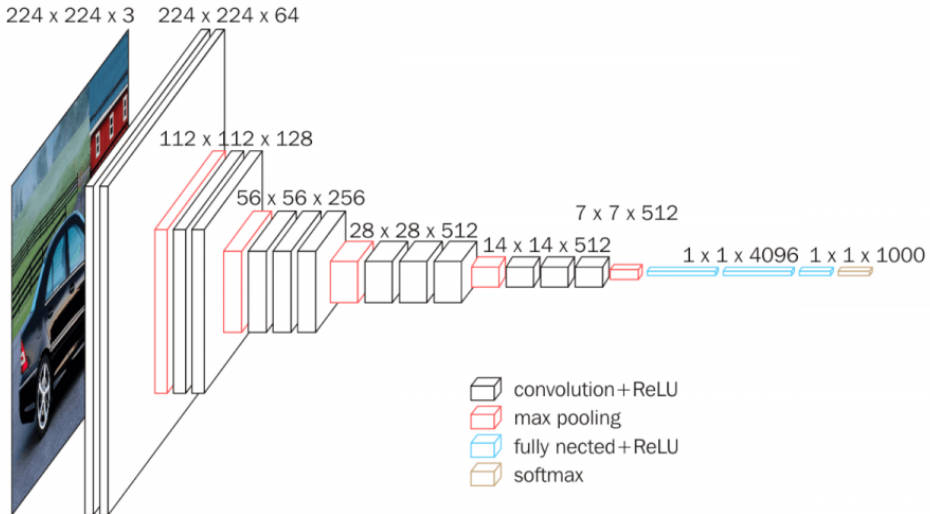


Fig. 1.3. VGG deep convolutional neural network architecture by Simonyan and Zisserman (2015)

GoogLeNet (also known as Inception-v1) and Inception-v3 are based on the idea of Inception modules, as illustrated in Fig. 1.4. The authors of Inception modules decided to give up fully-connected layers and instead stack many Inception modules throughout the network. This allows the model to adaptively choose between multiple convolutional filter sizes in each block.

The authors of ResNet introduced residual blocks, as illustrated in Fig. 1.5, which learn residual functions with respect to the layer inputs instead of learning unreferenced functions. This helps to overcome the problem of vanishing gradients and allows for the training of CNNs with a much larger number of layers. The original paper describes an architecture with up to 152 layers, which is a significant increase compared to its predecessors. Like VGG, this results in an improvement in quality but also a decrease in efficiency, both in terms of inference time and memory usage.

As an extension of the idea of using CNN features as the initial input to a language model, attention to these features was introduced by Xu et al. (2015).

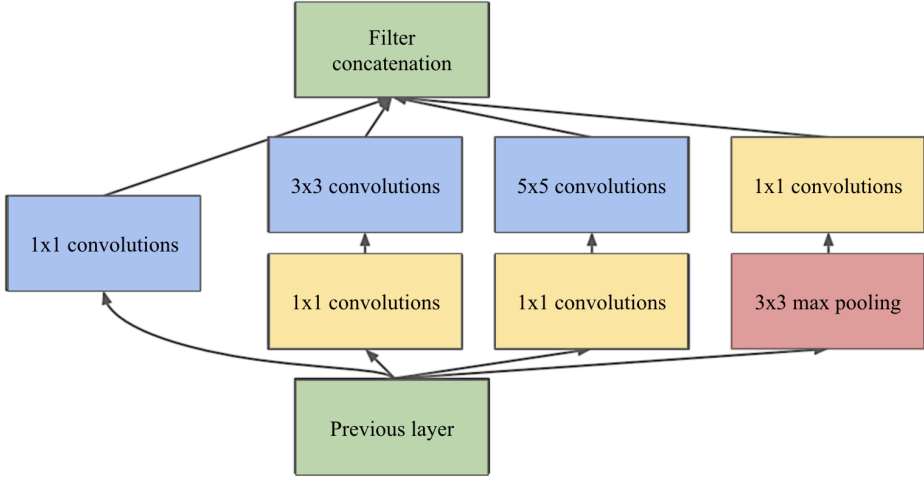


Fig. 1.4. Inception block introduced by Szegedy et al. (2015)

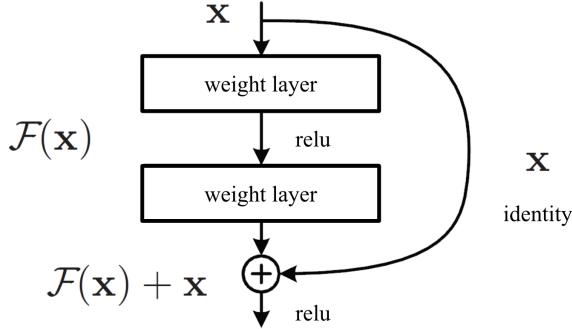


Fig. 1.5. Residual block introduced by He et al. (2016)

Suppose that the encoder model (such as previously described VGG or ResNet) encodes an image as a set of L vectors, each of which is D -dimensional:

$$a = a_1, \dots, a_L, a_i \in \mathbb{R}^D. \quad (1.4)$$

These vectors can be obtained using, for example, features from the lower convolutional layers of the networks. This representation is then used for the caption generation phase using the decoder.

For an LSTM network (which is often used as a decoder in such models), the process of generation based on the feature vectors is as follows. At each time step

t , not only the usual LSTM inputs, such as the hidden state or the output from the previous generation step), a context vector $\hat{\mathbf{z}}_t$ is used for conditioning. In general, the context vector is responsible for highlighting the exact features of the image that are important for generating the next word at that particular step.

To compute the context vector $\hat{\mathbf{z}}_t$, the soft attention method introduced by Bahdanau et al. (2015) is used. For each of the features in a at each time step t , the corresponding weight α_{ti} is calculated using the attention model f_{att} . The attention model itself can be as simple as a multilayer perceptron conditioned on the previous hidden state h_{t-1} . The attention model allows the calculation of the attention weights in the following way:

$$e_{ti} = f_{att}(\mathbf{a}_i, h_{t-1}). \quad (1.5)$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^L \exp(e_{tk})}. \quad (1.6)$$

The context vector $\hat{\mathbf{z}}_t$ is then computed as

$$\hat{\mathbf{z}}_t = \sum_{k=1}^L \alpha_{tk} \cdot \mathbf{a}_k. \quad (1.7)$$

Finally, the context vector is used as additional input to the LSTM network at time step t to generate the next word in the sentence. This allows the model to focus on different parts of the image at different times while generating the sentence, thus improving the quality of the generated captions.

The general encoder-decoder approach with an attention mechanism has been widely explored and modified in recent years. Lu et al. (2017) proposed using visual sentinels to make the attention mechanism adaptive and able to understand on the fly whether the next word generation requires attending to specific image regions or if it can rely solely on the language model. This is achieved by introducing a gate mechanism that controls the flow of information between the language model and the visual features. Wang et al. (2017) proposed a template-based method for generating captions. First, a “skeleton” template is generated using a neural network with attention, and then a description is generated by enriching the skeleton with words. In this approach, the skeleton-network attention is used to refine the attention for the description network. This allows the model to focus on the relevant parts of the image while generating the description. Ge et al. (2019) used attention in a bi-directional LSTM model, allowing the model to attend to both past and future words in the sentence while generating the current word. The models introduced by Gu et al. (2018) used attention at different levels of caption generation, from coarse

to fine. This allows the model to first generate a coarse caption with broad details and then refine it with increasing attention to specific details in the image. Yang et al. (2016) introduced a “review network” that performs a number of review steps using attention mechanisms over the encoder. This allows the model to generate more accurate and detailed captions by iteratively reviewing the generated words and attending to the relevant parts of the image. Chen et al. (2017) extended spatial attention, which is commonly used with channel-wise attention, by introducing the SCA-CNN model. This model uses separate attention mechanisms for spatial and channel-wise features, allowing for more fine-grained and accurate attention to the relevant image regions. Jiang et al. (2018) used multi-attention at different levels for fusing features from multiple visual encoders, rather than focusing on different parts of the input image. This allows the model to effectively combine information from multiple encoders and generate more accurate and descriptive captions. Finally, a number of research authors, e.g. Sugano and Bulling (2016), Tavakoli et al. (2017), Ramanishka et al. (2017), Cornia et al. (2018), and Chen and Zhao (2018) proposed using “human attention” to understand which areas of the image are important at each stage of caption generation. These methods integrate human gaze information into the attention over image features framework, allowing the model to generate captions that better match human descriptions of the same images.

However, even though the attention concept helps the model to understand which information extracted from the image is more and less important, during another step of sentence generation, all the features in described methods correspond to the image as a whole. Some of the later research efforts proposed using specific image regions for feature calculations instead. The difference between the two directions can be seen in Fig. 1.6. Earlier works used some handcrafted systems for region calculation. Thus, Fu et al. (2016) proposed using segmentation, then calculating hierarchy boxes on top and calculating patches of the image, which would be treated as different image regions. But later works usually follow the idea of using the image detector introduced by Anderson et al. (2018).

Even though the attention concept helps a model to understand which information extracted from an image is more and less important, during another step of sentence generation, all the features in described methods correspond to the image as a whole. Some of the later works propose to use specific image regions for feature calculations instead. This difference can be seen in Fig. 1.6, where earlier works use handcrafted systems for region calculation. For example, Fu et al. (2016) proposed using segmentation to calculate hierarchy boxes on top of the image, and then calculate patches of the image, which would be treated as different image regions.

Later works, however, usually follow the idea of using an image detector introduced by Anderson et al. (2018), who proposed that attention could vary and be

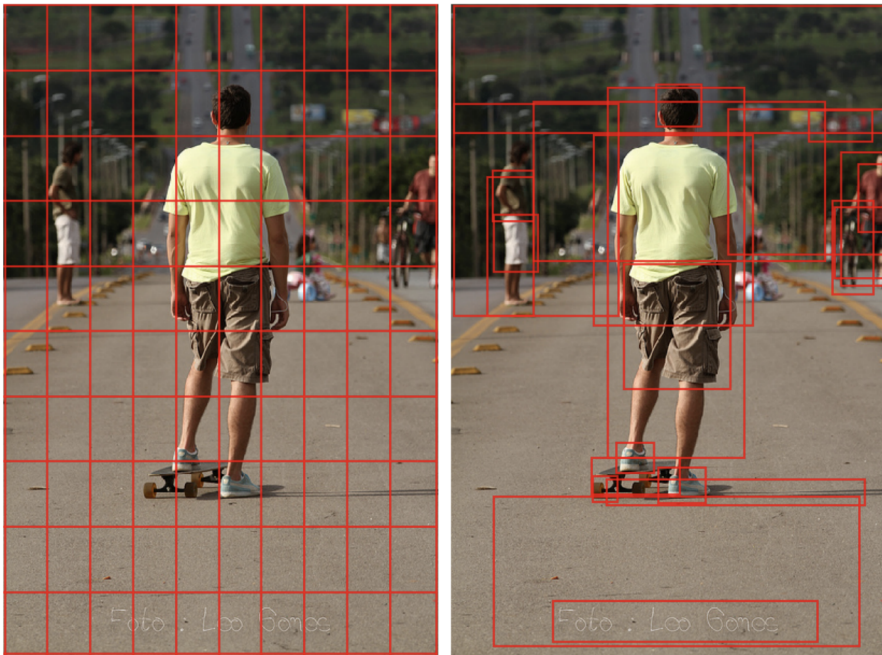


Fig. 1.6. Typical image subdivision for attention model prior to Anderson et al. (2018) (left) and subdivision introduced in Anderson et al. (2018) and called “bottom-up attention” (right)

not only of a “top-down” type (depending on the task that is currently being solved, such as the generation of the next word in a caption) but also “bottom-up” (based on unexpected signals). Top-down attention is typically represented through attention modules in a language model and was described above. Bottom-up attention is proposed to be introduced through an object detector on the image. The detection model called Faster R-CNN by Ren et al. (2015) is used in the paper to extract several image regions corresponding to the objects and then pass their features to the decoder as input for caption generation instead of passing the whole image features.

Apart from these methods, other approaches have been used for extracting image regions and then applying an attention module, such as described by Huang et al. (2019b), Ke et al. (2019), Wang et al. (2020), and Qin et al. (2019). However, these approaches have not become popular among researchers. In contrast, methods that use the CLIP model introduced by Radford et al. (2021) to calculate embeddings of detected objects (for example, by Shen et al. (2021)) are widely studied.

The next step to a higher level of abstraction was made by Yao et al. (2018) and extended in paper by Guo et al. (2019), where the authors claimed that apart from objects, the relations between them are important for caption generation. Thus, taking these relations as additional input during the encoding phase can improve the quality of the image captioning system. The authors introduced the Graph Convolutional Network (GCN) to capture both semantic and spatial information for encoding purposes. The overall system using GCN can be found in Fig. 1.7. Alternatively, Yang et al. (2019a) introduced the Scene Graph Auto-Encoder (SGAE) to model the relationships between objects to take advantage of the inductive bias inherent in human language.

Shi et al. (2020) described a multistage approach for incorporating visual relations. The authors use weakly supervised multi-instance learning to construct caption-guided visual relationship graphs, which are then improved with neighboring and contextual nodes. The authors also use multi-task learning to predict not only the caption but also the tag sequence for generator training. Hierarchy Parsing (HP) architecture was used for similar purposes by Yao et al. (2019), where it combines both instance-level and object-level information for the creation of a relations graph.

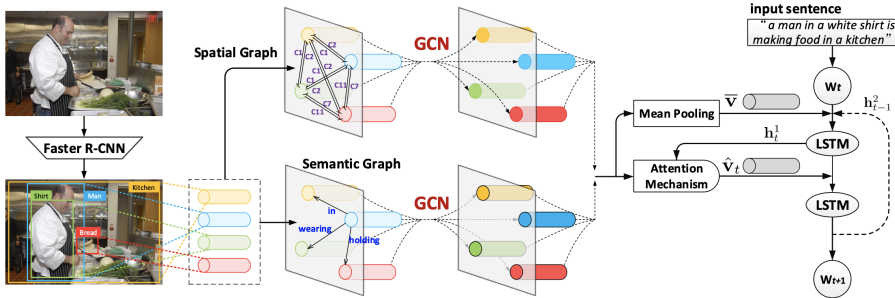


Fig. 1.7. Image captioning system using Graph Convolutional Network introduced by Yao et al. (2018)

Attempts to construct visual relationship graphs manually using heuristic algorithms have their natural limit due to their dependency on human knowledge. That is why when self-attention was first introduced by Vaswani et al. (2017) and then adopted not only for natural language processing but also for computer vision tasks, it quickly appeared in image captioning papers. One of the first works with this approach by Yang et al. (2019b) used self-attention for a relation module to seek interaction among the features in the Collocate Neural Model (CNM). Li et al. (2019a) used EnTangled Attention (ETA) to apply attention to both visual and semantic information.

Similarly, to incorporate geometrical information, Guo et al. (2020) proposed a different type of attention called Geometry-aware Self-Attention (GSA). The “Attention on Attention” network by Huang et al. (2019a) improves the conventional attention mechanism by calculating the similarity between attention results and queries. Another type of attention block called X-Linear is described by Pan et al. (2020). Its difference from the original attention module is the use of the bilinear pooling technique. In contrast Cornia et al. (2020b) solved the problem of limited attention abilities by adding additional “memory slots” for self-attention that can store some a priori information. The Image Transformer model, which can also make use of semantic graphs for attention, is used by He et al. (2020a). Another original approach can be found in the paper by Liu et al. (2020b), where the model is trained to predict the attention values for the next generated word, not the previous one (based on the previous hidden vector like in all the other works).

Ji et al. (2021) extended the attention idea to more than just the image region level, also using global image information. The Dual-Level Collaborative Transformer (DLCT) by Luo et al. (2021) is created to use both intrinsic object properties with Dualway Self Attention and geometric information with Comprehensive Relation Attention. Attempts to incorporate semantic information were also made by Cornia et al. (2020a) and Liu et al. (2019a). Zhang et al. (2021b) tried to retain more spatial information by not only considering the detected objects but also the image grid information for self-attention modules. Another researcher approach is to represent an image as a sequence and use the transformer architecture typically used for “sequence-to-sequence” tasks, as described by Liu et al. (2021).

1.1.5. Language Models

Language models are a crucial part of any image captioning system. Their main goal is to estimate the probability of a sequence of words occurring in natural language. Based on this model, a word generator can be built, which is suitable for the decoder part of an image captioning system. Therefore, it is not surprising that typical language model architectures are used for the decoder in image captioning tasks.

Formally, a language model P predicts the probability of a given sequence of words, optionally using some condition. In the case of image captioning, the condition is represented by a visual encoding of the image, which was discussed in detail in the previous sub-chapter. Therefore, the probability of a caption $C = (c_1, c_2, \dots, c_n)$ for a given image I can be formulated as:

$$P(c_1, c_2, \dots, c_n | I) = \prod_{i=1}^n P(c_i | c_1, \dots, c_{i-1}, I). \quad (1.8)$$

Once a trained language model P is available, a caption is typically gener-

ated step-by-step. Each subsequent word c_i is generated based on the estimated probability $P(c_i|c_1, \dots, c_{i-1}, I)$. In the simplest case, this is done using argmax:

$$c_i = \arg \max_{c_i \in vocab} P(c_i|c_1, \dots, c_{i-1}, I). \quad (1.9)$$

Another method is to use the beam search described in 1.1.

Initially, vanilla recurrent neural networks (RNNs) were used as language model decoders for image captioning systems. However, after the introduction of long-short term memory networks (LSTMs) by Hochreiter and Schmidhuber (1997), they quickly gained popularity and started to be used almost in all image captioning research. However, some other variants for a decoder architecture also exist.

A recurrent neural network (RNN) is a type of neural network that is well-suited to modeling sequential data, such as time series or natural language. RNNs are composed of multiple recurrent layers, each of which contains a hidden state that can capture information about the previous inputs in the sequence. Formally, let x_t be the input at time step t , and h_t be the hidden state at time step t . Then, the hidden state at time step $t + 1$ can be computed as follows:

$$h_{t+1} = f(W_{xh}x_t + W_{hh}h_t + b_h), \quad (1.10)$$

where f is a non-linear activation function, W_{xh} and W_{hh} are weight matrices, and b_h is a bias vector. The output at time step t can then be computed as:

$$y_t = g(W_{hy}h_t + b_y), \quad (1.11)$$

where g is another non-linear activation function, and W_{hy} and b_y are output weight matrix and bias vector, respectively.

Long short-term memory (LSTM) networks are a type of recurrent neural network that was introduced by Hochreiter and Schmidhuber (1997). LSTMs are designed to overcome the vanishing gradient problem, which is a common issue in RNNs when training on long sequences. This is achieved by introducing additional gates, which allow the LSTM to control the flow of information through the network. Formally, let i_t , f_t , o_t , and g_t be the input gate, forget gate, output gate, and input modulation gate, respectively, at time step t . Then, the hidden state at time step $t + 1$ can be computed as follows:

$$h_{t+1} = f_t \odot h_t + i_t \odot g_t, \quad (1.12)$$

where $\odot$ is the element-wise multiplication operator. This equation combines the previous hidden state h_t with the current input i_t and input modulation g_t using the forget gate f_t to control the flow of information.

And the output at time step t can then be computed as:

$$y_t = o_t \odot \tanh(C_t), \quad (1.13)$$

where C_t is the memory cell at time step t , which is computed as:

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t. \quad (1.14)$$

The gates are computed using the following equations:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}C_{t-1} + b_i), \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}C_{t-1} + b_f), \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}C_{t-1} + b_o), \\ g_t &= \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g), \end{aligned} \quad (1.15)$$

where σ is the sigmoid function, and W_{xi} , W_{hi} , W_{ci} , W_{xf} , W_{hf} , W_{cf} , W_{xo} , W_{ho} , W_{co} , W_{xg} , W_{hg} , b_i , b_f , b_o , and b_g are weight matrices and bias vectors.

LSTMs are generally considered to be better as decoders for image captioning than RNNs for several reasons. First, LSTMs can capture long-term dependencies in sequential data more effectively than RNNs. This is particularly important in image captioning tasks, where the generated captions often contain multiple words that depend on each other. Second, LSTMs are less prone to the vanishing gradient problem, which is a common issue in RNNs when training on long sequences. This allows LSTMs to be trained more effectively on image captioning tasks, where the sequences can be quite long. Finally, LSTMs are generally more computationally efficient than RNNs, which can be important when working with large datasets. All these factors make LSTMs well-suited for use as decoders in image captioning systems.

One variation of LSTMs used in language models for image captioning is the use of a visual sentinel, as proposed by Lu et al. (2017). In this approach, a separate neural network called the visual sentinel is trained to predict when the LSTM should attend to a specific part of the image. This allows the LSTM to focus on the most relevant regions of the image when generating the corresponding caption, improving the accuracy and descriptive power of the generated captions. Another variation of LSTMs used in language models for image captioning is the use of hidden state reconstruction, as proposed by Chen et al. (2018) and Ge et al. (2019). In this approach, the LSTM is trained to reconstruct the hidden state of the network at previous timesteps, using the current input and hidden state. This regularization technique helps to improve the performance of the LSTM by encouraging it to incorporate more contextual information into the generated captions. A third variation of LSTMs used in language models for image captioning is the use of multi-stage generation, as proposed by Wang et al. (2017) and Gu et al.

(2018). In this approach, the LSTM is trained to generate captions in multiple stages, starting with a coarse-grained description of the image and progressively refining the generated text to produce a more detailed and accurate caption. This approach allows the LSTM to incorporate both global and local contexts when generating the captions, improving the performance of the generated text.

Another variation of LSTMs used in language models for image captioning is the use of semantic-guided LSTMs, as proposed by Jia et al. (2015). In this approach, the LSTM is trained to generate captions based on a semantic representation of the image, rather than the raw pixel data. This allows the LSTM to incorporate more high-level information about the image content when generating the captions, improving the performance of the generated text.

Another common method found in many papers and used to improve language models for image captioning is two-layer LSTMs. Two-layer LSTMs, also known as stacked LSTMs, consist of two LSTM layers stacked on top of each other, with the output of the first layer serving as input to the second layer. This allows the network to incorporate more complex and abstract representations of the input sequence, improving the performance of the generated captions. One example of a two-layer LSTM used in image captioning is the “Neural Baby Talk” model proposed by Lu et al. (2018). In this model, the first LSTM layer encodes the input image into a set of visual features, while the second LSTM layer uses these features to generate the corresponding caption. A second example of a two-layer LSTM used in image captioning is the “Look back and predict forward” model proposed by Qin et al. (2019). In this model, the first LSTM layer encodes the input image into a set of visual features, while the second LSTM layer uses these features to generate the corresponding caption. Another example of a two-layer LSTM used in image captioning is the “Adaptive attention time” model proposed by Huang et al. (2019b). In this model, the first LSTM layer encodes the input image into a set of visual features, while the second LSTM layer uses these features to generate the corresponding caption.

The Transformer model, first proposed by Vaswani et al. (2017), has revolutionized the field of natural language processing (NLP). This architecture, which uses a fully-attentive paradigm, has since been used in many language understanding tasks, such as BERT Devlin et al. (2019). It has also been applied to image captioning, which can be formulated as a sequence-to-sequence problem. In the standard Transformer architecture for image captioning, a decoder performs a masked self-attention operation on words, followed by a cross-attention operation where words act as queries and the outputs of the last encoder layer act as keys and values. This is followed by a final feed-forward network. During training, a masking mechanism is used to enforce a unidirectional generation process. Several variants of the Transformer decoder have been proposed to improve language generation and visual feature encoding in image captioning. Some image captioning models, e.g. de-

scribed by Herdade et al. (2019) and Wang et al. (2021), have employed the original Transformer decoder without making significant changes to the architecture.

With the recent emergence of large language models, such as ChatGPT (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023), they could be widely used for image captioning systems' improvement. For example, ChatGPT is a language model that has been trained on a large corpus of text data using unsupervised learning techniques. It can generate coherent and contextually appropriate responses to user inputs. ChatGPT can be used to improve image captioning systems by generating more accurate and comprehensive captions for images.

ChatGPT can be used to generate captions that are more descriptive and informative than those generated by traditional image captioning systems. It can incorporate contextual information from the surrounding text and generate captions that are more nuanced and accurate. Additionally, ChatGPT can be used to generate captions that are more fluent and natural-sounding, which can improve user experience.

A comparison of the main image captioning models described in this sub-chapter can be found in Table 1.1.

Table 1.1. Comparison of the main encoder-decoder image captioning models. B@4 corresponds to BLEU-4, M corresponds to METEOR, R corresponds to ROUGE-L, C corresponds to CIDEr, and S corresponds to SPICE

Model	B@4	M	R	C	S
Show and Tell (Vinyals et al., 2015)	27.7	23.7	-	85.5	-
Show, Attend and Tell (Xu et al., 2015)	25.0	23.0	-	-	-
Up-Down (Anderson et al., 2018)	36.3	27.7	56.9	120.1	21.4
GCN-LSTM (Yao et al., 2018)	38.3	28.6	58.5	128.7	22.1
G+ (Guo et al., 2019)	38.4	28.5	58.4	128.6	22.0
SGAE (Yang et al., 2019a)	39.0	28.4	58.9	129.1	22.2
ETA (Li et al., 2019a)	39.9	28.9	59.0	127.6	22.6
AoANet (Huang et al., 2019a)	38.9	29.2	58.8	129.8	22.4
X-LAN (Pan et al., 2020)	39.5	29.5	59.2	132.0	23.4
M^2 (Cornia et al., 2020b)	39.1	29.2	58.6	131.2	22.6
GET (Ji et al., 2021)	39.5	29.3	58.9	131.6	22.8
RSTNet (Zhang et al., 2021b)	40.1	29.8	59.5	135.6	23.3
ORT (Herdade et al., 2019)	38.6	28.7	58.4	128.3	22.6

1.1.6. Joint Visual-Language Models

One of the key benefits of using vision-and-language pre-training for image captioning is that it can help the model to learn a strong initial representation of the

relationship between visual and language information. This can be particularly useful in the context of image captioning, where the goal is to generate a natural language description of an image. By pre-training the model on a large dataset of image-caption pairs, the model can learn to associate specific visual features and objects with their corresponding semantic concepts in the captions. This can improve the model's ability to generate accurate and descriptive captions for a given image.

Several different approaches exist to vision-and-language pre-training for image captioning. In the case of the Lxmert model by Tan and Bansal (2019), the authors use a transformer architecture to combine visual and language information from the outset. This allows the model to learn the relationship between visual and language modalities in a more integrated way, rather than treating them as separate input streams. The use of self-supervised pre-training on a large dataset of image-caption pairs is another key aspect of the Lxmert model. This allows the model to learn a strong initial representation of the relationship between visual and language information, which can improve performance on the image captioning task.

In contrast, the Oscar model by Li et al. (2020) uses a different approach to pre-training, known as object-semantic alignment. This involves pre-training the model on a large dataset of image-caption pairs, using a self-supervised learning objective that encourages the model to align the visual features of objects with the corresponding semantic concepts in the captions. By learning to associate specific visual features with their corresponding semantic concepts, the model can learn a more precise representation of the relationship between visual and language information. This can improve the model's ability to generate accurate and descriptive captions for a given image.

Zhou et al. (2020) presented the model called UNITER. The key feature of the UNITER model is its use of cross-modal contrastive learning. This involves using a contrastive loss function to encourage the model to generate more accurate and discriminative visual and language representations.

VinVL model presented by Zhang et al. (2021a) uses a variable-length fusion mechanism, which allows the model to better handle images with varying numbers of objects and captions with varying lengths. This allows the model to generate more diverse and flexible captions, improving performance on the image captioning task. In a later study, Hu et al. (2021) improved the VinVL model by scaling up its size and using larger-scale noisy data for pre-training. This allowed the model to better handle a wider range of input data and to generate more accurate and detailed captions.

Overall, early fusion and vision-and-language pre-training are important techniques that can be used for the task of image captioning. By combining visual and language information from the start and pre-training the model on a large dataset, these techniques can improve the performance of image captioning models

A comparison of the image captioning models based on joint visual-language models described in this sub-chapter can be found in Table 1.2. It is worth noting that though joint models could provide better alignment between internal representation of both language and vision modalities and have higher quality, their work is much harder in terms of explainability of generated captions. The other disadvantage of such models is their big size, which is to cover multimodal dependencies inside one model.

Table 1.2. Comparison of the main joint vision-language image captioning models. B@4 corresponds to BLEU-4, M corresponds to METEOR, C corresponds to CIDEr, and S corresponds to SPICE

Model	B@4	M	C	S
OSCAR (Li et al., 2020)	41.7	30.6	140.0	24.5
Unified (Zhou et al., 2020)	36.5	28.4	116.9	21.2
VinVL(Zhang et al., 2021a)	41.0	31.1	140.9	25.2
LEMON (Hu et al., 2021)	42.6	31.4	145.5	25.5

1.1.7. Generative Adversarial Networks for Image Captioning

Generative Adversarial Networks (GANs) are a class of machine learning models that have recently gained popularity in the field of computer vision and natural language processing. GANs can be used to generate realistic captions that are indistinguishable from real sentences. This section will discuss how GANs can be used for image captioning models.

GANs consist of two neural networks: a generator and a discriminator. The generator network is trained to generate captions that are indistinguishable from real captions. The discriminator network is trained to distinguish between real captions and fake captions generated by the generator network. The two networks are trained in an adversarial manner, with the goal of achieving a balance between generating realistic captions and detecting fake captions.

There are several ways to use GANs for different image captioning tasks. For example, Yan et al. (2018) used them to compensate for the exposure bias problem of maximum likelihood estimation. Unlike previously mentioned authors, Chen et al. (2019) dealt with the inconsistent evaluation problem among different objective language metrics using a discriminator. Wang and Cook (2020) proposed to use of bidirectional GAN to improve the quality of generated captions. Zhenxian et al. (2021) used GANs to generate more accurate and diverse captions.

Overall, GANs could help in image captioning applications as a discriminator network can be used to detect the mismatch between the image and the generated sentence.

1.2. Image Captioning Model Training and Evaluation Methods

One important aspect of training image captioning models is the selection of appropriate loss functions and training strategies. Loss functions are used to measure the difference between the model's predicted output and the ground truth labels. They are used to update the model's parameters during training.

One common loss function used in image captioning is cross-entropy loss. This loss is a common loss function used in supervised learning tasks, including image captioning. In image captioning, the goal is to train a model to generate natural language captions for a given image. The cross-entropy loss function is used to measure the difference between the model's predicted captions and the ground truth captions. It is used to update the model's parameters during training.

The cross-entropy loss function is calculated as the negative log likelihood of the true labels, given the predicted probabilities. This loss function can be written as:

$$L = - \sum_{i=1}^N y_i \log \hat{y}_i \quad (1.16)$$

where y_i is the ground truth label for the i -th example, and $\hat{y}_i$ is the predicted probability for the i -th example.

In image captioning, the ground truth labels are the true captions for the given images, and the predicted probabilities are the probabilities of the predicted captions. The cross-entropy loss function measures the difference between the predicted and ground truth captions using the cross-entropy between the two distributions.

The use of cross-entropy loss in image captioning allows the model to learn to associate specific visual features and objects with their corresponding semantic concepts in the captions.

During training, the cross-entropy loss is calculated for each predicted caption and the model's parameters are updated to minimize the loss. This can be done using gradient descent or other optimization algorithms. As the model is trained, the cross-entropy loss should decrease, indicating that the model is learning to generate more accurate captions.

The other loss that is usually used for an image captioning task is based on a reinforcement learning algorithm called REINFORCE, introduced by Williams (1992). The REINFORCE algorithm is a reinforcement learning technique that can be used for sequence generation tasks. The basic idea behind the algorithm is to learn a policy that can generate a sequence of actions that will maximize a certain reward. The algorithm works by using a parameterized policy function,

which takes in a sequence of observations and produces a sequence of actions. The policy is updated using gradient ascent on the expected reward, with the gradient being estimated using the Monte Carlo method. The algorithm has been shown to be effective for a variety of sequence generation tasks, including language modeling and machine translation.

The first attempt to incorporate REINFORCE algorithm for image captioning task was made by Ranzato et al. (2016), adopting BLEU by Papineni et al. (2002) and ROUGE by Lin and Och (2004) as reward signals. But the most widely used approach was introduced by Rennie et al. (2017). The typical setup using this loss is the following. The model is first trained to generate a sequence of words (i.e., a caption) for a given image using a standard supervised learning approach using cross-entropy loss described above. Then, the model's generated captions are evaluated using the reward function (CIDEr metric by Vedantam et al. (2015)), and the model is updated to generate captions that are expected to maximize the reward. This process is repeated until the model can generate high-quality captions for images.

One key aspect of self-critical sequence training is that the model is trained to optimize the expected reward of its own generated sequences rather than the ground-truth sequences. This allows the model to learn to generate sequences that are expected to maximize the reward rather than simply reproducing the training data. This can lead to improved performance on the image captioning task, as the model is able to generate more fluent and relevant captions for images.

Even though the self-critical sequence training approach is the most common method for training image captioning models using reinforcement learning, there have been some other attempts to develop alternative methods. For example, Ren et al. (2017) proposed a method that uses a reinforcement learning approach to train an image captioning model by incorporating an embedding reward signal. Liu et al. (2017) introduced a policy gradient optimization approach that uses a SPIDER-based reward signal for image captioning. Zhang et al. (2017) introduced an actor-critic sequence training approach for image captioning. Gao et al. (2019) introduced a self-critical n-step training approach, which uses a sequence-level reward signal that considers the long-term quality of the generated captions. These papers demonstrate some of the alternative methods that have been proposed for training image captioning models using reinforcement learning.

Overall, the selection of loss functions for image captioning models can have a significant impact on the performance of the model. Different loss functions can be used depending on the training strategy and the desired level of performance. Careful consideration of the available data and the desired performance characteristics can help to guide the selection of the most appropriate loss function.

1.2.1. Image Captioning Datasets

To train and evaluate image captioning models, it is necessary to have access to large-scale datasets that contain a diverse range of images and their corresponding captions.

One of the most common datasets for image captioning is the Microsoft COCO (Common Objects in Context) dataset by Lin et al. (2014). MS COCO (common objects in context) is a large-scale dataset that is commonly used for the task of image captioning. It contains a total of over 200,000 images, each annotated with multiple captions that describe the objects and scenes present in the images. The dataset is designed to be challenging, with a wide variety of objects and scenes, including complex and uncommon objects, as well as a diverse set of image backgrounds and contexts. The captions in the dataset are written in natural language, providing a rich source of data for training and evaluating image captioning models.

The images in the MS COCO dataset are organized into a several different categories, including common objects, scenes, and activities, allowing for a wide range of applications. In addition to the images and captions, the dataset also includes a several additional annotations, such as object bounding boxes and segmentation masks, which can be used to train and evaluate more sophisticated image captioning models.

The Flickr30k and Flickr8k datasets are other popular datasets used for the task of image captioning. They are both collections of images, each annotated with multiple natural language captions that describe the objects and scenes present in the images.

The Flickr30k dataset contains a total of 30,000 images, each annotated with five captions. The images in the dataset are diverse, with a wide range of objects and scenes, and are drawn from a variety of sources, including personal photographs and stock images. The captions in the dataset are written in natural language, providing a rich source of data for training and evaluating image captioning models.

The Flickr8k dataset is similar to the Flickr30k dataset, but it is smaller in size, containing only 8,000 images. Each image in the dataset is annotated with five captions, and the images are drawn from a variety of sources, including personal photographs and stock images. As with the Flickr30k dataset, the captions in the Flickr8k dataset are written in natural language, providing a rich source of data for training and evaluating image captioning models.

In addition to these datasets, there are also several other datasets that have been used for image captioning, including the Visual Genome dataset by Krishna et al. (2017) and the Pascal Sentences dataset. The Visual Genome dataset is a large-scale dataset that has been specifically designed for the task of image captioning. It contains a total of over 108,000 images, each annotated with a rich set of structured visual annotations, including object bounding boxes, object and scene attributes,

and relationships between objects. In addition to these annotations, the dataset also includes several natural language captions for each image. The Pascal Sentences dataset is a collection of images that have been specifically annotated with natural language sentences for the task of image captioning. It contains a total of over 100,000 images, each annotated with at least five different sentences. The images in the dataset are drawn from a variety of sources, including personal photographs and stock images, and the sentences are written in natural language, providing a valuable source of data for training and evaluating image captioning models.

Overall, the availability of large-scale datasets is crucial for the development of effective image captioning models. These datasets provide the necessary training data for the models and allow for the evaluation of the model's performance on a variety of images and captions.

1.2.2. Image Captioning Metrics

There are several metrics that are commonly used to evaluate the performance of image captioning models. These metrics are designed to measure the quality of the generated captions, and can be used to compare the performance of different algorithms and models.

The BLEU (bilingual evaluation understudy) score is a commonly used metric for evaluating the performance of image captioning models. This metric measures the overlap level between the generated caption and a set of reference captions using n-gram precision.

To calculate the BLEU score, the generated caption and the reference captions are first tokenized, splitting the text into individual words or phrases. The n-gram precision is then calculated for each value of n, from 1 to 4. This calculation involves counting the number of n-grams that appear in both the generated caption and the reference captions and dividing by the total number of n-grams in the generated caption. The n-gram precisions for each value of n are then combined using a weighted geometric mean to calculate the overall BLEU score:

$$BLEU = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right), \quad (1.17)$$

where BP is the brevity penalty, N is the maximum value of n, w_n is the weight for the n-gram precision, p_n is the n-gram precision for a given value of n, and $\log$ is the natural logarithm.

Higher BLEU scores indicate that the generated caption is more similar to the reference captions and is, therefore, considered to be of higher quality. The BLEU score of 1.0 indicates that the generated caption is identical to the reference captions, while a score of 0 indicates that there is no overlap between the generated caption

and the reference captions. In practice, a BLEU score is often used as a relative measure, with different algorithms and models being compared to one another based on their BLEU scores.

The ROUGE (recall-oriented understudy for gisting evaluation) score is another commonly used metric for evaluating the performance of image captioning models. This metric measures the level of overlap between the generated caption and a set of reference captions, using a variety of different metrics.

To calculate the ROUGE score, the generated caption and the reference captions are first tokenized, splitting the text into individual words or phrases. The ROUGE score is then calculated using several different metrics, including the ROUGE-N, ROUGE-L, and ROUGE-W metrics.

The ROUGE-N metric measures the overlap between the generated caption and the reference captions using n-gram precision and recall. This metric is calculated as follows:

$$ROUGE - N = \frac{\sum_{n=1}^N \text{count}(n\text{-grams in both})}{\sum_{n=1}^N \text{count}(n\text{-grams in candidate})}, \quad (1.18)$$

where N is the maximum value of n , and count is a function that counts the number of n-grams in the given input.

The ROUGE-L metric measures the overlap between the generated caption and the reference captions using a longest common subsequence (LCS) algorithm. This metric is calculated as follows:

$$ROUGE - L = \frac{\sum_{n=1}^N \text{LCS}(n\text{-grams in candidate}, n\text{-grams in reference})}{\sum_{n=1}^N \text{count}(n\text{-grams in candidate})}. \quad (1.19)$$

The METEOR (metric for the evaluation of translation with explicit ordering) score is a commonly used metric for evaluating the performance of image captioning models. This metric is a combination of the BLEU and ROUGE scores, and is designed to provide a more comprehensive measure of the quality of the generated captions.

To calculate the METEOR score, the generated caption and the reference captions are first tokenized, splitting the text into individual words or phrases. The BLEU score is then calculated using the n-gram precision and recall, as described in the previous answer. The ROUGE score is then calculated using several different metrics, including the ROUGE-N, ROUGE-L, and ROUGE-W metrics.

The METEOR score is then calculated as a weighted average of the BLEU and ROUGE scores, with the weights being determined based on the performance of the generated caption on several different tasks. The METEOR score is calculated

as follows:

$$METEOR = w_1 \cdot BLEU + w_2 \cdot ROUGE, \quad (1.20)$$

where w_1 and w_2 are the weights for the BLEU and ROUGE scores, respectively, and BLEU and ROUGE are the corresponding scores for the generated caption.

Higher METEOR scores indicate that the generated caption is of higher quality and is more similar to the reference captions in terms of both n-gram precision and recall and longest common subsequence. The METEOR score of 1.0 indicates that the generated caption is identical to the reference captions, while a score of 0 indicates that there is no overlap between the generated caption and the reference captions. In practice, the METEOR score is often used as a relative measure, with different algorithms and models being compared to one another based on their METEOR scores.

However, CIDEr (consensus-based image description evaluation) and SPICE (semantic propositional image caption evaluation) are two of the most common metrics for evaluating the performance of image captioning models. One of the key reasons why CIDEr and SPICE are popular metrics for image captioning is that they have been shown to be more effective at evaluating the quality of the generated captions than other metrics, such as BLEU and ROUGE. For example, Vedantam et al. (2015) found CIDEr to be significantly better than BLEU and ROUGE at evaluating the quality of image captions generated by state-of-the-art algorithms.

Another reason why CIDEr and SPICE are popular metrics for image captioning is that they can capture a wider range of aspects of the generated captions than other metrics. For example, SPICE can evaluate the semantic content of the generated captions, as well as their syntactic structure, whereas other metrics, such as BLEU and ROUGE, are only able to evaluate the syntactic structure of the captions. This ability to capture a wider range of aspects of the generated captions makes CIDEr and SPICE more effective at evaluating the overall quality of the captions.

To calculate the CIDEr score, the generated caption and the reference captions are first tokenized, splitting the text into individual words or phrases. The n-gram precision is then calculated for each value of n, from 1 to 4, as in the BLEU score. The n-gram precisions are then weighted using an inverse document frequency (IDF) scheme, which assigns higher weights to n-grams that appear less frequently in the reference captions. The weighted n-gram precisions are then combined using a weighted geometric mean to calculate the overall CIDEr score.

Formally, the CIDEr score is calculated using the following formula:

$$CIDEr = \frac{1}{N} \sum_{i=1}^N IDF(R_i) \cdot \left(\frac{\sum_{j=1}^{m_i} \text{sim}(r_{i,j}, \hat{r}_{i,j})}{m_i} \right). \quad (1.21)$$

Where:

- N is the number of images in the dataset;
- R_i is the reference caption for the image i ;
- m_i is the number of words in the reference caption for the image i ;
- $r_{i,j}$ is the j -th word in the reference caption for the image i ;
- $\hat{r}_{i,j}$ is the j -th word in the predicted caption for the image i ;
- $\text{sim}(r_{i,j}, \hat{r}_{i,j})$ is the similarity between the j -th word in the reference and predicted captions for the image i ;
- $IDF(R_i)$ is the inverse document frequency for the reference caption for the image i .

The similarity between two words is typically calculated using a modified version of the BLEU metric. The inverse document frequency is calculated as:

$$IDF(R_i) = \log \left(\frac{N}{n_i} \right), \quad (1.22)$$

where:

- N is the total number of images in the dataset;
- n_i is the number of images with the same reference caption as the image i .

Higher CIDEr scores indicate that the generated caption is more similar to the reference captions and is, therefore, considered to be of higher quality. The CIDEr score of 1.0 indicates that the generated caption is identical to the reference captions, while a score of 0 indicates that there is no overlap between the generated caption and the reference captions. In practice, the CIDEr score is often used as a relative measure, with different algorithms and models being compared to one another based on their CIDEr scores.

The SPICE (Semantic Propositional Image Caption Evaluation) measures the semantic correctness and fluency of the generated captions by comparing them to a set of reference captions for the same image. The metric is based on a set of predefined syntactic and semantic rules that define how the generated caption should match the reference caption in terms of content and structure.

The SPICE score is calculated using the following formula:

$$SPICE = \frac{1}{N} \sum_{i=1}^N \text{SPICE}(R_i, \hat{R}_i), \quad (1.23)$$

where:

- N is the number of images in the dataset;
- R_i is the reference caption for the image i ;

- $\hat{R}_i$ is the predicted caption for the image i ;
- $\text{SPICE}(R_i, \hat{R}_i)$ is the SPICE score for the reference and predicted captions for the image i .

The SPICE score for a pair of reference and predicted captions is calculated using the following formula:

$$\text{SPICE}(R_i, \hat{R}_i) = \frac{1}{|S_i|} \sum_{s \in S_i} F(s), \quad (1.24)$$

where:

- S_i is the set of all segments in the reference and predicted captions for image i ;
- $|S_i|$ is the number of segments in the set S_i ;
- $F(s)$ is the F-score for the segment s .

The F-score for a segment is calculated using the following formula:

$$F(s) = \frac{2 \cdot \text{Precision}(s) \cdot \text{Recall}(s)}{\text{Precision}(s) + \text{Recall}(s)}, \quad (1.25)$$

where:

- $\text{Precision}(s)$ is the precision for the segment s ;
- $\text{Recall}(s)$ is the recall for the segment s .

The precision and recall for a segment are calculated using the following formulas:

$$\text{Precision}(s) = \frac{|\text{TP}(s)|}{|\text{TP}(s)| + |\text{FP}(s)|}, \quad (1.26)$$

$$\text{Recall}(s) = \frac{|\text{TP}(s)|}{|\text{TP}(s)| + |\text{FN}(s)|}, \quad (1.27)$$

where:

- $\text{TP}(s)$ is the set of true positive propositions for the segment s ;
- $\text{FP}(s)$ is the set of false positive propositions for the segment s ;
- $\text{FN}(s)$ is the set of false negative propositions for the segment s .

The SPICE score ranges from 0 to 1, with higher values indicating the better performance of the image captioning model.

Overall, these metrics provide a way to quantitatively evaluate the performance of image captioning models and are commonly used to compare the performance of different algorithms and models.

1.2.3. Image Captioning Task Variations

Apart from the classical image captioning task, which involves generating a brief description of an image, there are several variations that have been explored by researchers. Some of these variations include dense image captioning, which involves generating a detailed and comprehensive description of an image; novel object captioning, which involves generating descriptions of objects that are not seen in the training data; personalized captioning, which involves generating descriptions that are tailored to a specific individual or group of individuals; and controllable captioning, which involves generating descriptions with specific attributes or attributes that can be controlled by the user. These variations of the image captioning task have the potential to be useful in a variety of applications and continue to be an active area of research.

Dense image captioning is a type of image description task that involves generating detailed and comprehensive descriptions of images by concurrently localizing and describing salient regions within the image. This is a more complex task than traditional image captioning, which only describes the whole image, or object detection, which only identifies and labels objects in the image. To tackle this task, researchers have proposed several different approaches, including Kim et al. (2019), which uses a triple-stream network to better capture the complex relationships between objects in an image, Krause et al. (2017), which uses a hierarchical model to generate a high-level summary of the image and then expand on it, Liang et al. (2017), which uses a generative adversarial network to generate captions with multiple sentences, each focused on a specific topic, Mao et al. (2018), which uses a topic-oriented model to generate multiple sentences for each image, each focused on a specific topic, Chatterjee and Schwing (2018), which uses a model trained to generate a diverse set of coherent captions, and Luo et al. (2019), which uses reinforcement learning to encourage the model to explore and generate diverse captions. These approaches aim to better capture the complex relationships between objects in an image and generate more accurate and detailed captions.

Novel object captioning is a task in natural language processing that involves generating descriptions of objects that are not present in the training data. This means that the model must be able to generalize its understanding of objects and their characteristics and use this knowledge to generate appropriate and accurate descriptions of novel objects. This is a challenging task, as it requires the model to have a strong understanding of the visual world and the language used to describe it. Approaches to this task are described by Hendricks et al. (2016), Venugopalan et al. (2017), Agrawal et al. (2019), Yao et al. (2017), Li et al. (2019c), Wu et al. (2018).

Personalized image captioning is a challenging task in natural language processing that involves generating descriptions of images that are tailored to a specific individual or group of individuals. This task requires the model to have a deep

understanding of the visual world, the language used to describe it, and the unique preferences and characteristics of the individual or group for whom the captions are being generated. To accomplish this, various approaches have been researched, e.g., by Chunseong Park et al. (2017), presenting a model that combines a generative neural network with a context sequence memory network to generate personalized image captions. Park et al. (2018) presented a model that uses a multimodal memory network to generate personalized image captions. Gan et al. (2017) and Shuster et al. (2019) explored the use of neural networks and incorporated concepts, such as “style” and “personality” in the caption generation process, resulting in captions that not only accurately describe the content of the image, but also use words and phrases that are aesthetically pleasing and reflect the personality of the speaker.

Controllable image captioning is a subfield of image captioning that focuses on generating captions that can be controlled or manipulated based on certain attributes or aspects of the image, such as depicted objects, scenes, or actions. For example, a user might want to generate a caption for an image that specifically describes the objects in the image without mentioning any background details or context. One significant challenge in the development of controllable image captioning models is the lack of large-scale datasets that include annotated control attributes. This makes it difficult to train and evaluate the performance of these models, as they require a large amount of data to learn from to generate high-quality captions. To address this issue, some researchers have proposed methods for automatically generating control attributes from existing image captioning datasets.

Some papers are designated for controllable image captioning. Cornia et al. (2019) presented a novel approach to image captioning that can generate diverse descriptions by allowing both grounding and controllability. The proposed framework uses a recurrent architecture to generate captions grounded on image regions and allows for control over the generated captions using a control signal in the form of a sequence or set of image regions. Deshpande et al. (2019) proposed a method for generating both diverse and accurate captions by first predicting a summary of the image using part-of-speech tags and then generating the caption based on the summary. Zhong et al. (2020) introduced a method for image captioning that involves decomposing the scene graph of an image into a set of sub-graphs, each of which captures a semantic component of the image. A deep learning model is used to select important sub-graphs and decode each selected sub-graph into a target sentence, allowing for the generation of captions that are accurate, diverse, grounded, and controllable. Chen et al. (2021) proposed the use of a new control signal called verb-specific semantic roles (VSR) for controllable image captioning. VSR consists of a verb and a set of semantic roles, representing the targeted activity and the roles of entities involved in that activity. The proposed method involves training a model to identify and ground entities for each role, learning human-like descriptive semantic structures, and using a role-shift captioning model to generate

captions. This approach allows for more human-like control over the generated captions.

1.3. Image Captioning Practical Problems

From a practical perspective, there are several challenges that limit the widespread adoption of image captioning systems in day-to-day life. According to the literature review, some of the most important problems include the following:

- Limited accuracy.

Image captioning models face challenges in accurately describing the content of images, especially in cases where the images are complex or contain multiple objects. This can result in incorrect or misleading captions that can be problematic in such applications as image retrieval or assistive technologies for the visually impaired. Additionally, the accuracy of image captioning models can be impacted by the quality of the training data and the complexity of the model architecture.

- Limited diversity.

Many image captioning models tend to generate repetitive or generic captions, which can limit their usefulness in applications, such as storytelling or artistic expression. This lack of diversity can be attributed to the fact that image captioning models tend to focus on generating captions that are most likely given the image rather than exploring alternative options.

- Large model sizes.

Many state-of-the-art image captioning models are large and computationally expensive, making them difficult to deploy in real-world settings. This is especially true for applications that require real-time processing or low-power devices, such as smartphones. The large model sizes can also increase the time required for training and testing, making it more challenging to iterate and improve the models.

- Limited domain-specific knowledge.

Image captioning models trained on general image datasets may struggle to accurately describe images in specialized domains, such as medicine, engineering, or law. This is because these domains often require domain-specific knowledge and terminology, which may not be present in general image datasets. To overcome this challenge, specialized image datasets need to be created, and models need to be trained using domain-specific knowledge.

1.4. Neural Network Model Compression Techniques

This sub-chapter reviews efficient neural network architectures, mostly concentrating on convolutional neural networks for the image detection task as such network type is usually used as an encoder for image captioning models. It also describes typical widely-used neural network model compression techniques, such as pruning, quantization, and knowledge distillation.

Neural network model compression is a set of techniques for reducing the size and computational complexity of neural network models without sacrificing their performance on a given task. These techniques are important because they allow neural network models to be more efficiently deployed on resource-constrained devices, such as mobile phones or embedded systems.

Several methods for compressing neural network models can be broadly classified into four categories: use of more efficient architectures, parameter pruning, quantization, and knowledge distillation.

1.4.1. Efficient Architectures

Using more efficient architectures involves replacing the original architecture of the neural network model with a more efficient one that has the same or better performance on the given task. This can be achieved by using more efficient building blocks, such as depthwise separable convolutions instead of regular convolutions for convolutional neural networks. Other methods for improving the efficiency of neural network architectures include using skip connections, which can reduce the computational complexity of the model, or using lightweight models, which are designed to be fast and efficient. More efficient architectures can be applied to both the encoder and decoder in an image captioning model to improve its efficiency. For the encoder, this can involve using a more efficient vision model, such as MobileNet instead of a VGG or ResNet, or using network pruning to remove redundant connections and parameters from the model. For the decoder, more efficient architectures can be used, such as an LSTM with skip connections or a transformer model with a smaller number of layers.

One effective way to reduce the size of a neural network model is to use a more efficient architecture that contains fewer parameters but still performs well on the given task. For example, the task of detecting objects in images (models of which usually used as the encoder for image captioning), along with their classification and attributes, can be solved using efficient architectures, such as Faster R-CNN by Ren et al. (2015), which uses a Region Proposal Network (RPN) and combines it with Fast R-CNN by Girshick (2015), along with a convolutional neural network (VGG-16 by Simonyan and Zisserman (2015)) as the backbone. SSD by Liu et al. (2016) also uses a convolutional neural network as the backbone, and creates a set

of default boxes over different aspect ratios and scales to detect objects of interest. Another efficient architecture for object detection is RetinaNet by Lin et al. (2017), which is a small, dense detector that uses focal loss-oriented training to achieve good performance.

The choice of the convolutional neural network architecture is also important for efficient object detection. Over the years, various methods have been proposed to reduce the number of parameters and the size of the model, such as MobileNetV3 by Howard et al. (2019), which was found through a neural architecture search based on previous MobileNet versions. Another popular convolutional neural network architecture is EfficientNet by Tan and Le (2019), which was specifically designed to reduce the number of parameters without sacrificing performance. This network is also used in the paper on EfficientDet by Tan et al. (2020), which applies the ideas from EfficientNet to the task of object detection.

1.4.2. Neural Network Pruning Methods

Parameter pruning involves removing individual parameters (e.g., weights or biases) from the network that are considered unimportant based on some criterion. For example, parameters with small magnitudes may be considered unimportant and can be removed from the network without significantly impacting its performance, as described by Molchanov et al. (2019). The resulting network will have fewer parameters, which can reduce the amount of memory required to store the network and the computational resources required to perform inference with the network. Parameter pruning can also be applied to both the encoder and decoder in an image captioning model. For the encoder, this can involve identifying and removing the parameters that have the smallest impact on the model's performance or using regularization techniques that encourage the model to learn more compact representations. For the decoder, pruning can be used to remove unnecessary parameters from the language model, such as the attention weights or the hidden states of the LSTM.

Pruning is a widely-used method for reducing the size of neural network models. It was described by Hoefler et al. (2021), He et al. (2020b), and Tanaka et al. (2020). The idea behind pruning is to remove model parameters that are not useful or meaningful, without significantly degrading the model's performance on the given task. This can help to save storage space and computational resources while still maintaining a good level of model quality.

Pruning methods can be broadly divided into two categories: structured pruning (Anwar et al., 2017; He et al., 2018) and unstructured pruning (Lee et al., 2018; Xiao and Wang, 2019). Structured pruning involves removing entire rows, columns, or channels of weights from the model, which can be effective for reducing the size of convolutional neural network (CNN) models or models using recurrent neural networks (RNNs). Unstructured pruning, on the other hand, involves removing

individual weights from the model without considering the underlying structure of the model. This can be done based on the importance of each weight, as determined by a specific criterion. Unstructured pruning methods have been applied to a variety of tasks and can achieve good performance in many cases.

Mathematically, weight pruning can be represented as follows:

$$w \leftarrow w \odot m \quad (1.28)$$

Here, w is the original weights of the network, $\odot$ is the element-wise multiplication operator, and m is a binary mask that indicates weights that should be removed from the network (i.e., $m_i = 0$ for weights that should be removed, and $m_i = 1$ for weights that should be kept). The resulting weights w will have fewer non-zero elements than the original weights, which can reduce the number of parameters in the network.

1.4.3. Neural Network Quantization Methods

Quantization involves representing the parameters of the network using a lower-precision data type, such as 8-bit integers or 16-bit floating-point numbers, instead of the standard 32-bit floating-point numbers like in the paper by Courbariaux et al. (2016). This can reduce the amount of memory required to store the network, and can also enable the use of hardware accelerators that are optimized for low-precision data types, which can reduce the computational resources required to perform inference with the network. Quantization can be applied to both the encoder and decoder in an image captioning model to reduce the precision of the model's parameters. For the encoder, this can involve using lower precision data types, such as 8-bit integers instead of 32-bit floating point numbers, to reduce the size and computational complexity of the model. For the decoder, quantization can be used to reduce the precision of the language model's parameters, such as the word embeddings or the hidden states of the LSTM.

Neural network quantization (Gholami et al., 2021) is a method of compressing a model without losing quality. This is done by using lower precision numbers for storing the model, which reduces the amount of occupied space without affecting the model's quality. In most cases, the precision provided by standard floating point data types is not necessary for practical applications.

There are two main types of quantization: static and dynamic. In static quantization (Jacob et al., 2018; Yao et al., 2021), the range of values that will be compressed is calculated before the model is used and remains the same during runtime. In dynamic quantization (Choi et al., 2018; Li et al., 2019b), this range is calculated for each activation map during the model's use.

There are two main approaches to quantizing a neural network: post-training quantization and quantization-aware training. Post-training quantization involves applying a quantization function to the weights and activations of a pre-trained network. This function maps each value to a lower precision quantized value. For example, the uniform quantization function maps each value to the nearest value in a uniform grid of quantized values. This is given by the formula:

$$Q(x) = \text{round} \left(\frac{x}{s} \right) \cdot s \quad (1.29)$$

Where x is the input value, s is the quantization step size, and round is the rounding function.

Post-training quantization can be applied to weights and activations at different levels of precision, depending on the needs of the model. For example, full-precision quantization uses the same precision as the original model, while low-precision quantization uses a lower precision.

Quantization-aware training, on the other hand, involves training a network with quantization in mind from the beginning. This is done by using a quantization-aware training objective that considers the quantization error that will be introduced when the network is quantized. The resulting network is designed to be quantized to a lower precision without sacrificing accuracy.

One approach to quantization-aware training is to use a quantized version of the network during training and optimize the weights and activations to minimize the quantization error. This can be done using a quantization-aware training objective, such as the mean squared error between the full-precision and quantized versions of the network. This is given by the formula:

$$\text{MSE}(w, q) = \frac{1}{N} \sum_{i=1}^N (w_i - q_i)^2 \quad (1.30)$$

Here, w is the full-precision weight or activation, q is the quantized weight or activation, and N is the number of weights or activations in the network. The mean squared error measures the difference between the full-precision and quantized versions of the network and can be used as a training objective to minimize the quantization error.

Another approach to quantization-aware training is to use a full-precision network during training but simulate the effects of quantization by adding a quantization noise term to the weights and activations of the network. This allows the network to learn to be robust to quantization and can result in a network that is better suited for low-precision quantization.

1.4.4. Image Captioning Model Knowledge Distillation

Knowledge distillation is a method for transferring knowledge from a large, complex model (called the teacher) to a smaller, simpler model (called the student). This is done by training the student model to mimic the outputs of the teacher model, using a combination of the ground-truth labels and the teacher model's predicted probabilities as supervision. By using knowledge distillation, it is possible to train a small, efficient model that can match or even surpass the performance of the original, larger model. Knowledge distillation can be applied to the encoder and decoder in an image captioning model to transfer knowledge from a large, complex model to a smaller, more efficient one. For the encoder, this can involve training a small, lightweight vision model to mimic the outputs of a larger, more complex model, using the ground-truth labels and the teacher model's predicted probabilities as supervision. For the decoder, knowledge distillation can be used to train a small, efficient language model that can match or even surpass the performance of the original, larger model. By using these techniques, it is possible to train highly efficient image captioning models that can be easily deployed on resource-constrained devices.

Knowledge distillation is a method that could be described as follows. Suppose we have a training dataset and want to train some compact model on it. But instead of training it straight away on the training data, bigger models are trained first. It is called the "teacher". The initial compact model, called the "student" is trained to mimic the "teacher". It could be trained differently from trying to learn intermediate representations of the "teacher" model to training on the final "teacher" model outputs. Sequence-level knowledge distillation is a type of knowledge distillation technique for sequence generation tasks in which the "student" model is trained on the dataset generated by fully applying the "teacher" model. It differs from trying to force the "student" to generate every next word the same as the "teacher" model would generate.

In the context of sequence generation tasks, such as machine translation and text summarization, knowledge distillation can be used to transfer knowledge from a large, pre-trained model (the teacher) to a smaller model (the student) that has fewer parameters and can be more easily deployed on resource-constrained devices. The teacher model is trained on a large dataset to predict the next word in a sequence of words. The student model is then trained on the same dataset but uses the output of the teacher model as additional supervision in addition to the ground truth labels. This allows the student model to learn from the knowledge encoded in the teacher model, which can improve its performance on the sequence generation task.

In the case of machine translation, the student model can be trained using a combination of the cross-entropy loss between teacher and student predictions and the cross-entropy loss between the student predictions and the ground truth labels:

$$\mathcal{L} = (1 - \alpha) \cdot \text{CE}(y, \hat{y}) + \alpha \cdot \text{CE}(\hat{y}, \hat{\hat{y}}). \quad (1.31)$$

Here, y is the ground truth label, $\hat{y}$ is the prediction of the teacher model, $\hat{\hat{y}}$ is the prediction of the student model, and α is a hyperparameter that controls the trade-off between the two losses. The student model is trained to minimize this loss function, which allows it to learn from the knowledge encoded in the teacher model.

In addition to improving the performance of the student model, knowledge distillation can also reduce the computational resources required to perform inference with the student model. This is because the student model has fewer parameters than the teacher model, which means it requires less memory and computational power to make predictions.

1.4.5. Image Captioning Model Compression Methods

Model compression methods are another important topic in the image captioning field. Most state-of-the-art image captioning methods are based on heavy memory-consuming neural networks that are unsuitable for practical application. Thus, for example, according to the section 1.1.4 most models use VGG16 or ResNet101 as encoders (or at least as backbones for image detectors) with weights equal to 530 megabytes and 170 megabytes, respectively. More than that, as models typically consist not only of an encoder but also a decoder part, it could also have a weight of the comparable scale. It leads to such consequences that most state-of-the-art models cannot be used on memory-restricted devices. A comparison of memory required to store the models for different types of state-of-the-art methods is presented in Table 1.3. According to the table, all the models that are not specially designed to be small have weights not less than 600 MB making them unsuitable for edge devices.

Image captioning model compression methods are not very well studied in the literature. There are only a few papers investigating this topic.

Rampal and Mohanty (2020) investigated a novel method for decreasing the sizes of an encoder and a decoder in an end-to-end pipeline. However, tests only focused on several models built on an outdated architectural approach (without object detection as an encoder). Furthermore, no extensive evaluation was undertaken of the main image captioning metrics, such as CIDEr and SPICE, which makes it hard to reasonably compare with the other methods. Tan et al. (2022) concentrated on the novel pruning method called “Supermask Pruning”, which performs gradual sparsification based on weight sensitivity to the training loss. The paper concentrated only on the pruning method for model compression and mostly on the decoder part, limiting the efficiency of such an approach.

Liu et al. (2020a) concentrated on knowledge distillation methods used particularly for vision-language pre-trained models. Fang et al. (2021) also focused

Table 1.3. Memory consumption of different image captioning models

Model	Size
Without pre-training	
Up-Down (Anderson et al., 2018)	661.4 MB
AoANet (Huang et al., 2019a)	791.8 MB
Normal model design	
VinVL (Zhang et al., 2021a)	954.4 MB
LEMON (Hu et al., 2021)	954.4 MB
Oscar (Li et al., 2020)	659.2 MB
VLP (Zhou et al., 2020)	659.2 MB
ViTCAP (Fang et al., 2022)	745.4 MB
BLIP (Li et al., 2022)	745.4 MB
Small model design	
DistillVLM (Liu et al., 2020a)	154.5 MB
MiniVLM (Fang et al., 2021)	154.5 MB
LightCap (Wang et al., 2022)	145 MB

on vision-language pre-trained models but mostly covered different decoders use aspects. Paper Wang et al. (2022) proposed completely new model architectures and concentrated mostly on the inference speed but not the memory aspect.

However, despite these efforts, the field of image captioning model compression is still understudied, and there is a lack of comprehensive studies on the effectiveness of different compression methods and their impact on model performance. However, there seems to be no research designated to different combinations of deep learning model compression techniques, such as hyperparameter change, pruning, quantization, and knowledge distillation for image captioning models.

1.5. Augmentation Techniques for Dataset Extension

Augmentation is a method of creating additional training data from an existing dataset. Many different augmentation techniques exist for various data types, and augmentation has been shown to be effective for tasks involving the analysis of such structured data as images (Wang and Perez, 2017) and text (Fadaee et al., 2017; Kobayashi, 2018; Zhang et al., 2015). Specific methods used for data augmentation often depend on the task at hand, but there are some general approaches that can be applied. For example, in the case of image augmentation (Wang and Perez, 2017), such techniques as horizontal flipping and random cropping are commonly used. These methods can be combined in various ways to create a larger and more diverse training set. In the case of text augmentation, such techniques as random word deletions and insertions, synonymous replacements are used (Zhang et al., 2015).

While augmentation has been widely used in tasks involving structured data, it is less commonly applied to vision-language tasks such as image captioning, visual question answering, and visual dialog. Most works in these areas, such as by Wang et al. (2016) and Wang et al. (2018), rely on standard image augmentation techniques. Others, e.g., Cui et al. (2018), use text augmentation techniques, such as word permutations and random word replacement, to improve the design of evaluation metrics. However, there have been some efforts to develop more sophisticated augmentation methods for vision-language tasks, for example, by Kafle et al. (2017). The authors used a template-based generation method that combines image annotations and an LSTM-based language model to generate question-answer pairs for images.

1.6. Conclusions of the First Chapter and Formulation of the Dissertation Tasks

The following conclusions can be made for this chapter:

1. One of the big, understudied problems of image captioning models is their massive memory consumption. This topic has a high practical interest; however, current models are too big to be used on memory-restricted devices.
2. In most papers, the same datasets (such as MS COCO and Flickr30k) are used with the same preprocessing techniques. However, the question of extending these datasets to have more than five different captions describing the same image remains open.
3. The effects of different encoder types on the quality of the model have not been extensively studied, although VGG and ResNet are the most common encoders used in the literature.
4. To improve the quality of language models used in image captioning, different approaches have been proposed. However, there is a lack of research on improving the internal representations of models, such as word embeddings.

Based on the conclusions, the following tasks are formulated to achieve the research goal:

1. To evaluate the possibility of using neural network compression methods to reduce the size of image captioning models without big loss of quality.
2. To propose and evaluate text augmentation methods for extending existing datasets and improving the quality of existing image captioning systems;
3. To evaluate the effect of changing the decoder in image captioning models.
4. To propose and evaluate the use of pre-trained word embeddings in image captioning models and compare it to training embeddings from scratch.

Reducing Image Captioning Neural Network Models' Memory Consumption

This chapter focuses on research on reducing image captioning neural network models size without significant loss of their metrics quality. Although most research efforts are concentrated on improving the quality of generated text regardless of a growing model size, this chapter shows that the size could be effectively reduced without any significant influence on metrics. Various methods were compared and used to reduce the size of the Up-Down and AoANet image captioning models. These methods include techniques applied to both model architectures, as well as methods that are specific to each individual model. It is important to note that the goal was not to compare the Up-Down and AoANet models directly but rather to evaluate the effectiveness of different model compression techniques on a range of models. A comparison of the original Up-Down and AoANet models can be found in the paper by Huang et al. (2019a).

This chapter starts by examining the results of encoder architecture changes. The use of several different architectures for the encoder part is proposed and compared with the most common model. It was found that the use of EfficientDet architecture could lead to a drastic reduction in memory consumption, still leaving quality metrics on a high level. The encoder part of the model size became more than 30 times smaller, while both CIDEr and SPICE decreased by no more than 1.5%.

Further, decoder architecture hyperparameter changes are proposed and evaluated. It has been shown that it allows for reducing the decoder's memory consumption by up to three times with only up to 1% CIDEr and SPICE metrics decrease.

In the following parts, the effect of decoder pruning techniques on model size and quality is researched. Using the best-found parameters of pruning leads to an even bigger size reduction compared to the initial one. Then, the quantization effect on the final model is evaluated. Finally, the knowledge distillation approach for model size reduction application is examined.

The experiment results were presented at an international conference and published in the proceedings (Atliha and Šešok, 2022a, 2022b).

2.1. Experiment Setup

All experiments in this chapter used the following evaluation setup.

MSCOCO by Lin et al. (2014) was used, which is the largest and most used dataset for image captioning a dataset for training, validating, and testing. Its standard version consists of 82,783 training images and 40,504 validation images. There are five different captions for each of the images. For offline evaluation, the standard Karpathy split by Karpathy and Fei-Fei (2015) was used, which is employed by most articles for result comparison. As a result, the final dataset consists of 113,287 images for training, 5,000 images for validation, and 5,000 images for testing.

Also postprocessing was performed by replacing all the words that occurred less than five times in the final dataset with a special token <UNK>. Furthermore, as vast majority of captions were no more than 16 words in length, words were truncated to maintain the maximal length of the caption equal to 16.

Widely used metrics such as BLEU, METEOR, ROUGE-L, CIDEr, and SPICE, were used to compare the efficiency of the methods, as well as metrics specific to the model compression task, such as memory consumption, number of parameters, and number of non-zero parameters.

A server with Intel(R) Xeon(R) CPU E5-2696 v4 @ 2.20GHz processor with 32 CPU cores, 120 GB RAM, Tesla V100 GPU card. Ubuntu 16.04 operation system was used as a computational device, and PyTorch and nlpaug libraries were installed.

2.2. Encoder Compression

The encoder is an essential component of most image captioning models. It is often the largest and heaviest part of the model, represented by a convolutional neural network. As such, compressing the encoder is one of the most effective ways to reduce the overall memory consumption of an image captioning model.

2.2.1. Encoders for Image Captioning

State-of-the-art image-captioning architectures often employ an image object detector as an encoder. This involves generating k image features $E(I) = V = v_1, v_2, \dots, v_k$, where each $v_i \in \mathbb{R}^d$ encodes a region of the input image I . The decoder then generates a caption $D(V)$ based on these features.

The use of this general scheme is proposed but with different encoders, which are more suitable for the specific task of the dissertation. One option is to retain the Faster R-CNN (Ren et al., 2015) detection model, but with a different backbone network that has fewer parameters and is better suited for memory-restricted tasks. The use of MobileNetV3 (Howard et al., 2019) is proposed as the backbone network for this purpose. Also, the using the EfficientDet detector (Tan et al., 2020) was considered, which is specifically designed for high performance with limited memory.

Therefore, the encoder options are Faster R-CNN ResNet101 (used in the original Up-Down model), Faster R-CNN MobileNetV2, and EfficientDet. Further encoder compression methods were not explored because these detectors already have relatively small sizes.

2.2.2. Empirical Evaluation and Results and Discussion of Empirical Evaluation of Encoder Choice

Tables 2.1 and 2.2 present the results of comparing Up-Down and AoANet models modified with different encoders. The variants with a Faster R-CNN ResNet101 encoder are the same as the models used in the original papers. As shown, all the proposed variations for both Up-Down and AoANet models have quality metrics that are very similar to each other. This aligns with the fact that, despite their small size, the image detection models used as encoders in these experiments show excellent results in standalone object detection tasks.

Furthermore, using smaller encoders designed for resource-constrained devices helps reduce the encoder's size from 468.4 MB to 15.1 MB (a 96.8% reduction), with only a slight decrease of 0.6 in CIDEr and 0.3 in SPICE scores. The EfficientDet-D0 encoder was determined as the most suitable for this task in the tested group and was used in all this chapter's experiments.

2.3. Decoder Compression

Since the current models may have more parameters than are necessary to achieve the same quality, first, the investigation determined whether the number of parameters could be reduced without changing the architecture. In this case, the method for

Table 2.1. Evaluation results of the main metrics of all tested models with different encoders. Both the “Size” and “Number of Parameters” columns refer to the encoder

Encoder Type	CIDEr	SPICE	Size	Number of Parameters
Up-Down				
Faster R-CNN ResNet101	120.1	21.4	468.4 MB	122.6 M
Faster R-CNN MobileNetV3	119.4	21	74.2 MB	19.4 M
EfficientDet-D0	119.5	21.1	15.1 MB	3.9 M
EfficientDet-D1	120	21.3	25.7 MB	6.6 M
AoANet				
Faster R-CNN ResNet101	129.8	22.4	468.4 MB	122.6 M
Faster R-CNN MobileNetV3	127.2	21.9	74.2 MB	19.4 M
EfficientDet-D0	129.2	22.1	15.1 MB	3.9 M
EfficientDet-D1	129.5	22.1	25.7 MB	6.6 M

Table 2.2. Evaluation results of the remaining metrics of all tested models with different encoders

Encoder Type	BLEU-1	BLEU-4	METEOR	ROUGE-L
Up-Down				
Faster R-CNN ResNet101	79.8	36.3	27.7	56.9
Faster R-CNN MobileNetV3	79.3	35.6	27.4	56.8
EfficientDet-D0	79.4	35.8	27.3	56.7
EfficientDet-D1	79.7	36.2	27.5	56.9
AoANet				
Faster R-CNN ResNet101	80.2	38.9	29.2	58.8
Faster R-CNN MobileNetV3	79.4	38	29.5	58.2
EfficientDet-D0	79.8	38.3	29.6	58.3
EfficientDet-D1	80.1	38.4	29.8	58.4

reducing the number of parameters depends on the specific model under study. Then pruning methods were applied to the best-performing architecture and, finally, quantized the best-pruned model.

2.3.1. Architecture Hyperparameter Changes for Compression

Following Anderson et al. (2018), the Up-Down decoder model has three important logical parts:

- Embeddings Calculation

In this part, one-hot encoded vectors representing words are transformed to word vectors. Let Π be a one-hot encoding of the input word, and $W_e \in \mathbb{R}^{E \times |\Sigma|}$ is a word-embedding matrix for a vocabulary Σ . Then, word

vector π could be obtained by the following equation:

$$\pi = W_e \Pi. \quad (2.1)$$

Here, the size of the parameter matrix depends on embedding dimension E and the size of the vocabulary $|\Sigma|$. As the size of the vocabulary is a fixed value based on the preprocessing of the dataset, we experimented with changing E and explored its influence on the overall model performance as well as on the model size.

- Top-Down Attention LSTM and Language LSTM

As these two modules are similar to each other in terms of the number of parameters and inner representations, we treat them as one logical part. Both modules are LSTMs, so generally, their operation over a single time step is the following:

$$h_t = \text{LSTM}(x_t, h_{t-1}), \quad (2.2)$$

where $h_t \in \mathbb{R}^M$. In this case, the sizes of the parameter matrices of LSTM strongly depend on parameter M . We manipulated this parameter by trying to reduce it without causing huge losses in the model's quality.

- Attention Module

The third important module of the Up-Down architecture is called Attention Module. It is used to calculate attention weights α_t for the encoder's features v_i . It works in the following way:

$$\alpha_{i,t} = w_a^T \tanh(W_{va} v_i + W_{ha} h_t^1), \quad (2.3)$$

$$\alpha_t = \text{softmax}(a_t), \quad (2.4)$$

where $W_{va} \in \mathbb{R}^{H \times V}$, $W_{ha} \in \mathbb{R}^{H \times M}$. The main constants that influence the amount of weights in this module are H , V , and M . V is the size of the encoder vectors, which is fixed by encoder choice, M has been discussed previously, and H is the dimension of input attention representation. We investigated how changing H influenced both model size and its performance along with other parameters.

Thus, in trying to reach a smaller model size without harming its performance metrics, we manipulated it with model parameters such as E , M , and H .

The model AoANet described by Huang et al. (2019a), similar to Up-Down, has both embedding calculation and LSTM parts. So, the reasoning for parameters E and M is similar to the previous subsection. However, the other parts of the architecture are different, so no experiments were performed with them.

The number of all the parameters was reduced by multiplying E , M , and H for Up-Down and E and M for AoANet used in the original papers by the equal scale factor γ .

2.3.2. Empirical Evaluation and Discussion of Architectural Hyperparameter Choice Results

Tables 2.3 and 2.4 evaluate models with their decoder reduced using different scale factors. Using a scale factor of $\gamma = 1$ corresponds to using the original models but with the EfficientDet-D0 encoder. For the Up-Down model and AoANet, reducing the decoder with a scale factor of 0.5 resulted in comparable results to the model without reduction but with approximately three times less memory and the number of parameters. However, further reducing the model leads to a significant decrease in quality. For example, the Up-Down model's CIDEr metric decreased by only 0.3 points when the scale factor of 0.5 was used, but when the scale factor of 0.25 was used, it decreased by 4.9 points. A similar situation was seen for the AoANet model's CIDEr score. Therefore, using a scale factor of $\gamma = 0.5$ is a good compromise that allows for a reduction in the model size while maintaining the same level of quality. This scale factor was used for experiments described in the rest of the chapter.

Table 2.3. Evaluation results of the main metrics of all tested models with different decoder scale factors. Both the “Size” and “Number of Parameters” columns refer to the decoder

Decoder Scale Factor	CIDEr	SPICE	Size	Number of Parameters
Up-Down				
$\gamma = 1$	119.5	21.1	193 MB	50.1 M
$\gamma = 0.5$	119.2	20.9	68.8 MB	18 M
$\gamma = 0.25$	114.3	19.8	27.5 MB	7.2 M
AoANet				
$\gamma = 1$	129.2	22.1	323.4 MB	84.8 M
$\gamma = 0.5$	129.1	22.1	100.7 MB	26.4 M
$\gamma = 0.25$	122.9	20.9	35.2 MB	9.2 M

2.3.3. Decoder Neural Network Pruning

This dissertation proposes to use unstructured pruning on the decoder after training has been completed. This method is appropriate because most of the layers of the considered decoders are linear or LSTM layers.

There are two parts of the decoder: calculating embeddings and processing them. The embedding part is a separate, important part because it is responsible

Table 2.4. Evaluation results of the remaining metrics of all tested models with different decoder scale factors

Decoder Scale Factor	BLEU-1	BLEU-4	METEOR	ROUGE-L
Up-Down				
$\gamma = 1$	79.4	35.8	27.3	56.7
$\gamma = \mathbf{0.5}$	79.4	35.2	27.1	56.6
$\gamma = 0.25$	78.3	33.2	26.2	55.5
AoANet				
$\gamma = 1$	79.8	38.3	29.6	58.3
$\gamma = \mathbf{0.5}$	79.5	38.5	29.5	58.3
$\gamma = 0.25$	78.5	36.4	28.6	57.4

for the selection of vectors that will represent the words. To determine the effect made by pruning of embeddings as the main semantic part on the quality of a model, two options are considered: pruning of the entire model and pruning of everything except embeddings.

Let the model to be pruned be $M(W_1, W_2)$, where $W_1 \in \mathbb{R}^{m_1}$ is a vector of model parameters that will not be pruned, and $W_2 \in \mathbb{R}^{m_2}$ is a vector of model parameters that will be pruned, sorted by the increase in their l_1 norms. Let $\alpha \in [0, 1]$ be the pruning coefficient, and $A(W, \alpha)$ be the pruning algorithm, which generates a mask of weights to be pruned. The final model, after pruning, will be $M(W_1, W_2 \cdot A(W_2, \alpha))$, where $\cdot$ denotes elementwise multiplication.

We concentrate on two methods of choosing weights to prune:

- For random pruning, $A(W)$ generates a mask $M \in \{0, 1\}^{|W|}$, where $M_i = 0$ with probability α .
- For l_1 pruning, $A(W)$ generates a mask $M \in \{0, 1\}^{|W|}$, where $M_i = 0 \forall i < \alpha|W|$.

2.3.4. Empirical Evaluation Results and Discussion for Decoder Pruning

Pruning results using different methods and pruning coefficients are presented in Figs. 2.1 and 2.2 for the Up-Down model and Figs. 2.3 and 2.4 for the AoANet model. Blue and orange lines on both figures correspond to l_1 pruning, while red and green lines are correspondent to random pruning. “True” represents pruning the embedding layer, and “False” represents not doing so.

It can be observed that random pruning consistently performs worse than l_1 pruning. This may be because removing less valuable weights from the model is expected to have a smaller impact on the model’s quality metrics than removing random parameters from the model. Additionally, not pruning the embeddings layer helps to maintain higher quality for larger pruning coefficients for the Up-Down

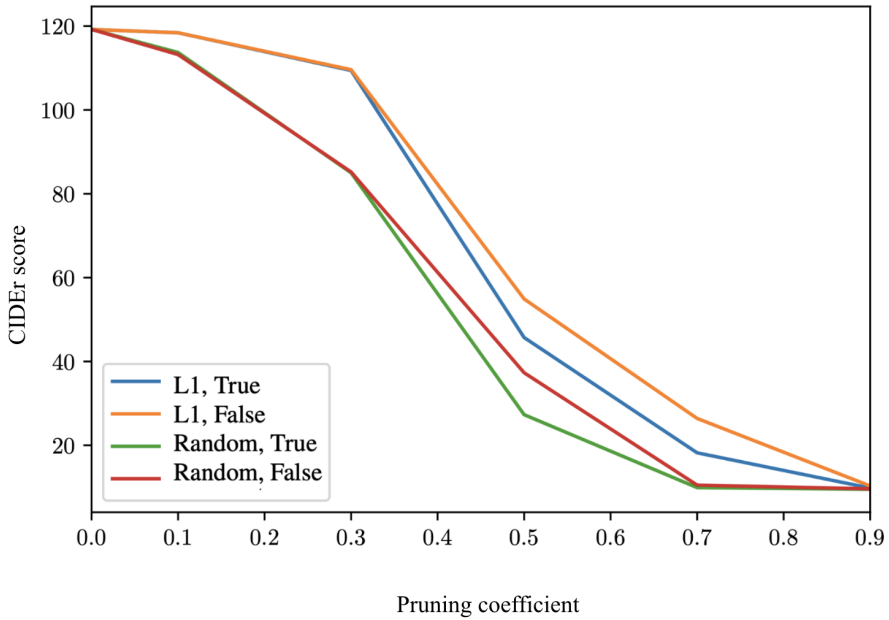


Fig. 2.1. CIDEr validation score dependence on pruning coefficient α for different pruning methods applied to the Up-Down model. “L1” indicates that l_1 pruning was used, while “Random” indicates that random pruning was used. “True” indicates pruning of the embedding layer, and “False” indicates no pruning

model but does not show much difference for the AoANet model. A significant decline in metrics can be observed for the Up-Down model starting from $\alpha = 0.1$ and for AoANet starting from $\alpha = 0.5$. Given this information, no interest is taken in the range of Up-Down model pruning where the choice of pruning or not pruning embeddings could make a difference.

As previously mentioned, the largest pruning coefficients (which result in the greatest model compression) that still produce good quality are 0.1 and 0.5 for the Up-Down and AoANet models, respectively. Increasing these values leads to a drop in metrics. More detailed comparisons of model results near this drop, along with the unpruned model quality, are reported in Tables 2.5 and 2.6. NNZ stands for “number of non-zero parameters”, which is a common measure for comparing pruning techniques. The table confirms the observations from the figures. The chosen boundary values for the pruning coefficients help to reduce the model size by 10% and lose only 0.6 CIDEr points for the Up-Down model, and reduce the model size by 50% and lose only 1.5 CIDEr points for the AoANet model. This large reduction in AoANet could indicate that the overall model is bigger and has

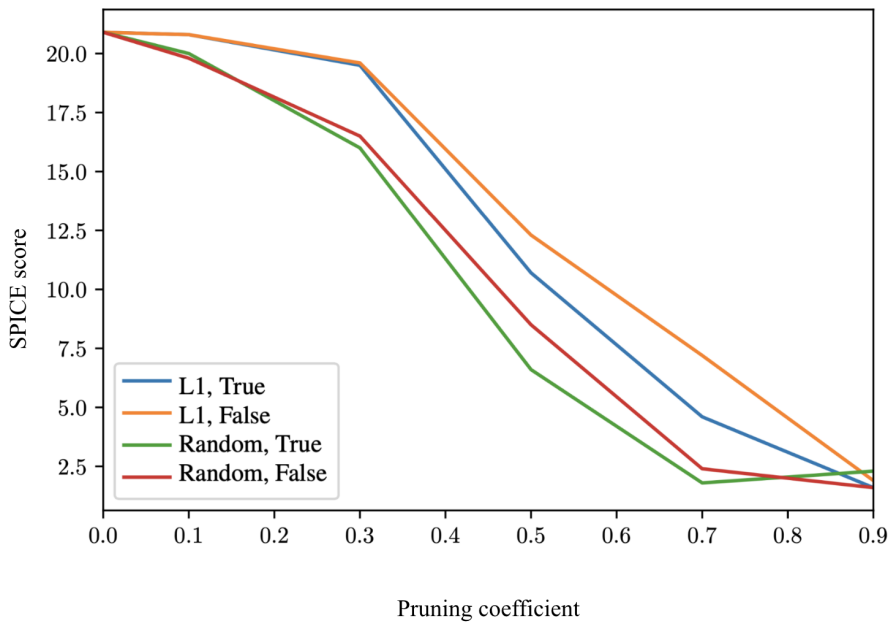


Fig. 2.2. SPICE validation score dependence on pruning coefficient α for different pruning methods applied to the Up-Down model. “L1” indicates that l_1 pruning was used, while “Random” indicates that random pruning was used. “True” indicates pruning of the embedding layer, and “False” indicates no pruning

more parameters than necessary to achieve comparable results with its architecture. Additionally, this model’s weights may be somewhat sparse, and many may be close to 0. In this case, such extensive pruning would not result in a significant loss in performance.

Table 2.5. Evaluation results of the main metrics of all tested model decoders pruned using different pruning coefficients. Both “Size” and “NNZ” refer to the decoder

Pruning Coefficient	CIDEr	SPICE	Size	NNZ
Up-Down				
$\alpha = 0$	119.2	20.9	68.8 MB	18 M
$\alpha = 0.1$	118.4	20.8	62 MB	16.2 M
$\alpha = 0.3$	109.4	19.5	48.2 MB	12.6 M
AoANet				
$\alpha = 0$	129.1	22.1	100.7 MB	26.4 M
$\alpha = 0.5$	127.6	21.5	50.4 MB	13.2 M
$\alpha = 0.7$	112	19	30.2 MB	7.9 M

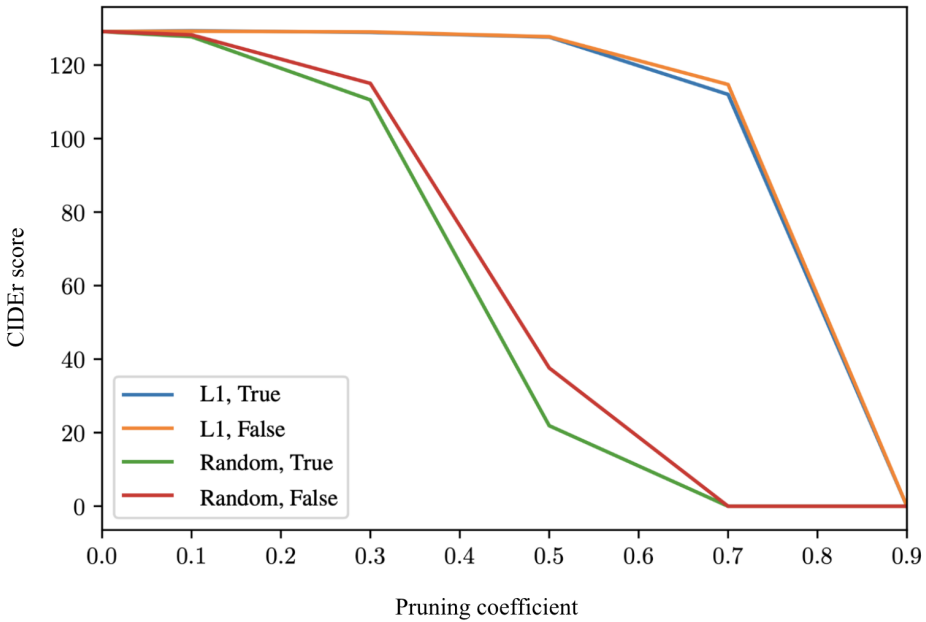


Fig. 2.3. CIDEr validation score dependence on pruning coefficient α for different pruning methods applied to the AoANet model. “L1” indicates that l_1 pruning was used, while “Random” indicates that random pruning was used. “True” indicates pruning of the embedding layer, and “False” indicates no pruning

2.3.5. Decoder Neural Network Quantization

To fix the model, post-training dynamic quantization was utilized. This method of quantization involves converting from floating point numbers to integers by multiplying by a constant and then rounding the result so that it fits into an int. The constant can be defined in different ways. The key benefit of dynamic quantization is that it allows for quantizing the weights of the model before applying it, which keeps the model compressed, while still calculating a constant by which to multiply activations during the calculation based on the input data. This approach enables maintaining the maximum possible accuracy while still storing the model in a compressed form. This method of quantization was chosen because of the primary interest in the accuracy and size of the model rather than the speed of its application.

For the quantization experiments, the best models obtained in the previous subsection were used, specifically, the Up-Down model pruned with $\alpha = 0.1$ and the AoANet model pruned with $\alpha = 0.5$.

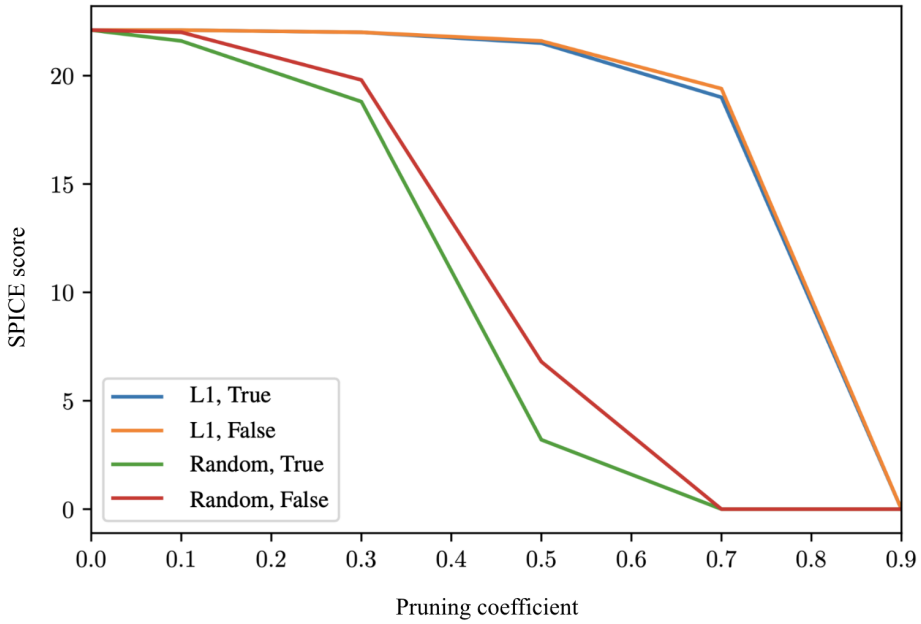


Fig. 2.4. SPICE validation score dependence on pruning coefficient α for different pruning methods applied to the AoANet model. “L1” indicates that l_1 pruning was used, while “Random” indicates that random pruning was used. “True” indicates pruning of the embedding layer, and “False” indicates no pruning

2.3.6. Empirical Evaluation Results and Discussion for Decoder Quantization

The results of the quantization experiments can be found in Tables 2.7 and 2.8. Quantization helps to reduce model size without approximately any loss of quality. The resulting decoder size of the Up-Down model is 43.4 MB, and the decoder size of the AoANet model is 19.7 MB.

A graphical representation of the final method can be found in Fig. 2.5.

2.4. Image Captioning Model Knowledge Distillation

In a series of experiments, the use of sequence-level knowledge distillation was explored as an alternative to pruning and quantization techniques for accelerating the quality of the models investigated in Sub-chapter 2.3.2. Models were trained with $\gamma = 0.5$ and $\gamma = 0.25$ using this method.

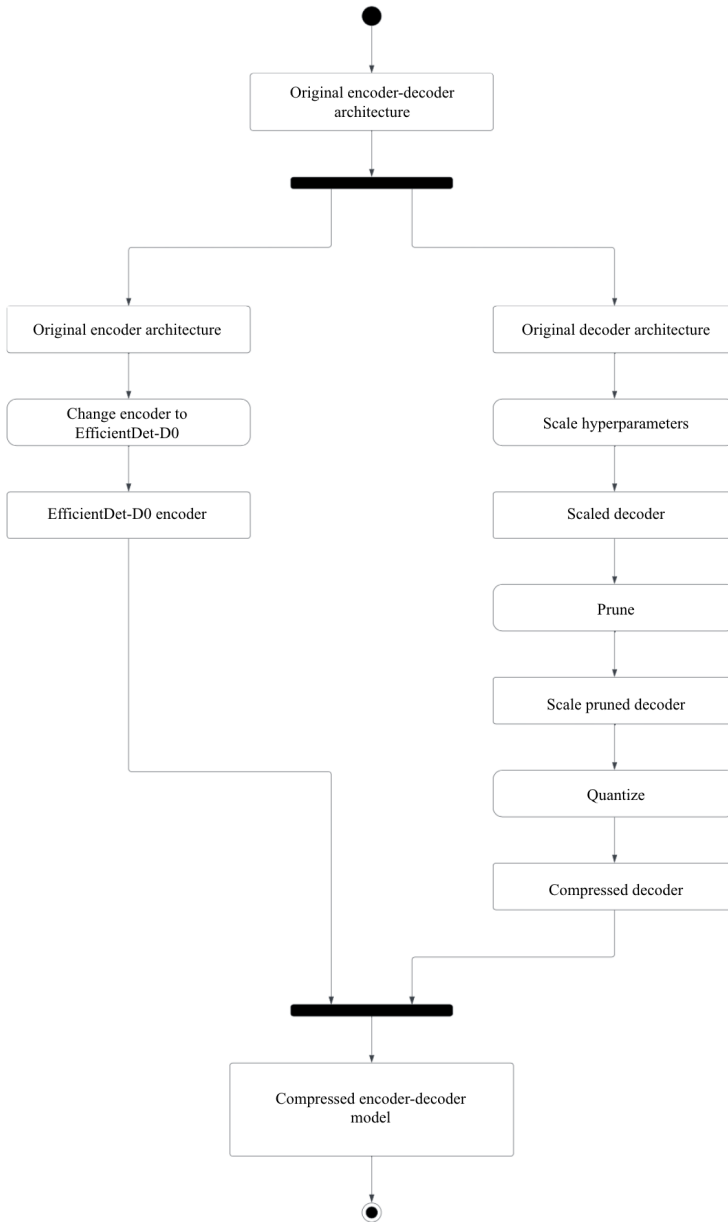


Fig. 2.5. Graphical representation of the first proposed compression method

Table 2.6. Evaluation results of the remaining metrics of all of the tested model decoders pruned using different pruning coefficients

Pruning Coefficient	BLEU-1	BLEU-4	METEOR	ROUGE-L
Up-Down				
$\alpha = 0$	79.4	35.2	27.1	56.6
$\alpha = 0.1$	79.2	35.1	26.9	56.3
$\alpha = 0.3$	77.5	32.6	25.7	54.7
AoANet				
$\alpha = 0$	79.5	38.5	29.5	58.3
$\alpha = 0.5$	79.3	38.2	29.1	58.3
$\alpha = 0.7$	75.5	34.1	26.6	56.3

Table 2.7. Evaluation results of the main metrics of all the tested model decoders with and without quantization. The “Size” column refers to the decoder

	CIDEr	SPICE	Size
Up-Down			
Without quantization	118.4	20.8	62 MB
With quantization	118.4	20.8	43.4 MB
AoANet			
Without quantization	127.6	21.5	50.4 MB
With quantization	127.4	21.4	19.7 MB

Table 2.8. Evaluation results of the rest metrics of all of the tested models decoders with and without quantization

	BLEU-1	BLEU-4	METEOR	ROUGE-L
Up-Down				
Without quantization	79.2	35.1	26.9	56.3
With quantization	79.2	35.3	26.9	56.3
AoANet				
Without quantization	79.5	38.2	29.1	58.3
With quantization	79.4	38.3	29.0	58.3

The proposed sequence-level knowledge distillation approach works as follows. Up-Down and AoANet models were used with $\gamma = 1$ (i.e., without decoder compression) as the “teacher” models. Then, new datasets were generated using these teacher models and used to train smaller models in the usual way.

To generate the dataset for training the smaller models, first, the teacher models were run on a set of input images to produce a set of corresponding output sequences. Then, these output sequences were used as the target sequences for the smaller models. This allowed the smaller models to learn from the higher-level knowledge of the teacher models, as represented by the output sequences.

2.4.1. Empirical Evaluation Results and Discussion of Knowledge Distillation

The potential influence of several parameters on the quality of sequence-level knowledge distillation was thoroughly examined. These parameters include:

- the beam size used for generating sequence with the “teacher” model;
- the size of the “student” model;
- the type of image captioning model (whether the “student” model is of the same but smaller architecture as the “teacher” model or whether it is a different architecture altogether);
- intermediate training (whether the small model is trained on the outputs of the big model or whether an intermediate-sized model is first trained as a “student” and then used as a “teacher”).

First, beam size influence on the knowledge distillation was investigated. Four different values were used for beam size: 1, 3, 5, and 10. The results of experiments with different beam sizes are shown in Tables 2.9 and 2.10. Beam size equal to “-” corresponds to a model trained on the original dataset.

Table 2.9. Beam size influence on knowledge distillation for models with decoder scale factor $\gamma = 0.5$

Beam Size	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Up-Down					
-	35.2	27.1	56.1	119.2	20.9
1	35.6	27.1	56.5	117.6	20.8
3	35.9	27.1	56.6	118.4	20.7
5	35.4	26.9	56.2	117.6	20.6
10	35.8	27.1	56.6	118.5	20.8
AoANet					
-	38.5	28.5	58.3	129.1	22.1
1	37.9	28.3	57.8	126.3	21.8
3	37.6	28.1	57.6	125.3	21.6
5	37.8	28.3	57.9	125.9	21.8
10	38	28.3	57.8	126.3	21.7

As demonstrated by the results, beam size has a discernible effect on the quality of knowledge distillation, although the relationship is not straightforward. For both Up-Down architecture models, the optimal beam size was found to be 10, while for the AoANet architecture, the situation was slightly different: for the model with $\gamma = 0.5$, performance was the same for beam sizes of 1 and 10, and for the model with $\gamma = 0.25$, the optimal beam size was 3. Nonetheless, it can be concluded that

Table 2.10. Beam size influence on knowledge distillation for models with decoder scale factor $\gamma = 0.25$

Beam Size	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Up-Down					
-	33.6	26.2	55.5	114.3	19.8
1	35.2	26.8	56.2	116	20.3
3	35.2	26.7	56.1	115.6	20.4
5	35.3	26.7	56.1	115.8	20.4
10	35.4	26.9	56.3	116.6	20.5
AoANet					
-	36.6	27.6	57.4	122.9	20.9
1	37.3	28	57.4	124.3	21.5
3	37.5	28	57.5	124.5	21.5
5	37.4	27.9	57.3	124.2	21.5
10	37.2	28	57.4	123.8	21.5

the potential benefit of using a knowledge distillation approach depends on the size of the model. In the case of models with $\gamma = 0.5$, knowledge distillation did not provide any improvement, but for models with $\gamma = 0.25$, it resulted in a significant boost, increasing the CIDEr metrics by 2.3 and 0.9 points, and the SPICE metrics by 0.7 and 0.6 points for Up-Down and AoANet models, respectively.

In the next stage of the study, the role of the teacher model architecture was ascertained in determining the final quality of the knowledge distillation process. To this end, Up-Down models were trained using AoANet as the teacher model, and vice versa, using AoANet models with Up-Down as the teacher. The results of these experiments are presented in Table 2.11. This analysis aimed to determine whether it is the architecture of the teacher model or its metrics that have the greatest impact on the quality of knowledge distillation.

Table 2.11. Architecture's influence on knowledge distillation for models

Decoder Scale Factor	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Up-Down					
$\gamma = 0.5$	36.6	27.3	57.2	119.7	20.9
$\gamma = 0.25$	35.3	27	56.3	116.8	20.6
AoANet					
$\gamma = 0.5$	37	28	57.1	123.5	21.7
$\gamma = 0.25$	37	27.8	57.3	122.8	21.4

The results of our experiments indicate that both Up-Down models showed an improvement in their metrics, while AoANet models exhibited a decrease. Given that the AoANet teacher model was initially superior to the Up-Down teacher model,

this outcome supports the notion that the quality of the teacher model is the primary factor in determining the quality of the knowledge distillation process. These findings suggest that using a high-quality teacher model is crucial for achieving good results in knowledge distillation.

The final experiment of the study aimed to determine the effect of intermediate training on the quality of knowledge distillation. To this end a model was trained with $\gamma = 0.25$ using a model with $\gamma = 0.5$ that had been trained on the same dataset as the initial teacher model and also trained a model with $\gamma = 0.25$ using a model with $\gamma = 0.5$ that had been distilled from the initial teacher model. These experiments aimed to determine whether intermediate training, where the small model is trained on a model with a similar number of parameters, can improve the performance of the small model. The results of these experiments are presented in Table 2.12.

Table 2.12. Intermediate training influence on knowledge distillation for models with decoder scale factor $\gamma = 0.25$ that use models with decoder scale factor $\gamma = 0.5$ as the teacher

Teacher Model Training	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Up-Down					
On original dataset	34.7	26.6	55.9	114.9	20.3
Distilled from a bigger model	35.2	26.8	56.2	115.5	20.3
AoANet					
On original dataset	37.3	27.9	57.2	123.4	21.3
Distilled from a bigger model	37	27.8	57.1	122.7	21.4

As shown in Table 2.12, intermediate training did not provide any benefit in any of the tested configurations. Since the quality of the medium-sized model was inferior to that of the larger model, it can be concluded that the size of the teacher model is not a determining factor in the quality of knowledge distillation and that only the quality of the teacher model is important. This suggests that it is essential to use a high-quality teacher model, regardless of its size, to achieve good results in knowledge distillation.

A graphical representation of the final method involving knowledge distillation can be found in Fig. 2.6.

In Tables 2.13 and 2.14, the final comparison between all proposed in the chapter compression methods is presented, as well as the comparison with methods by other authors specially designed to have a smaller model size. The “Original model” refers to the original model without any compression method applied. The “First method” is the best combination of encoder and decoder architectural changes, pruning, and quantization methods: EfficientDet-D0 encoder, decoder with a scale factor 0.5, pruning coefficients 0.1 and 0.5 for Up-Down and AoANet models, correspondingly, and quantization applied. “Second methods” is the best combination of encoder

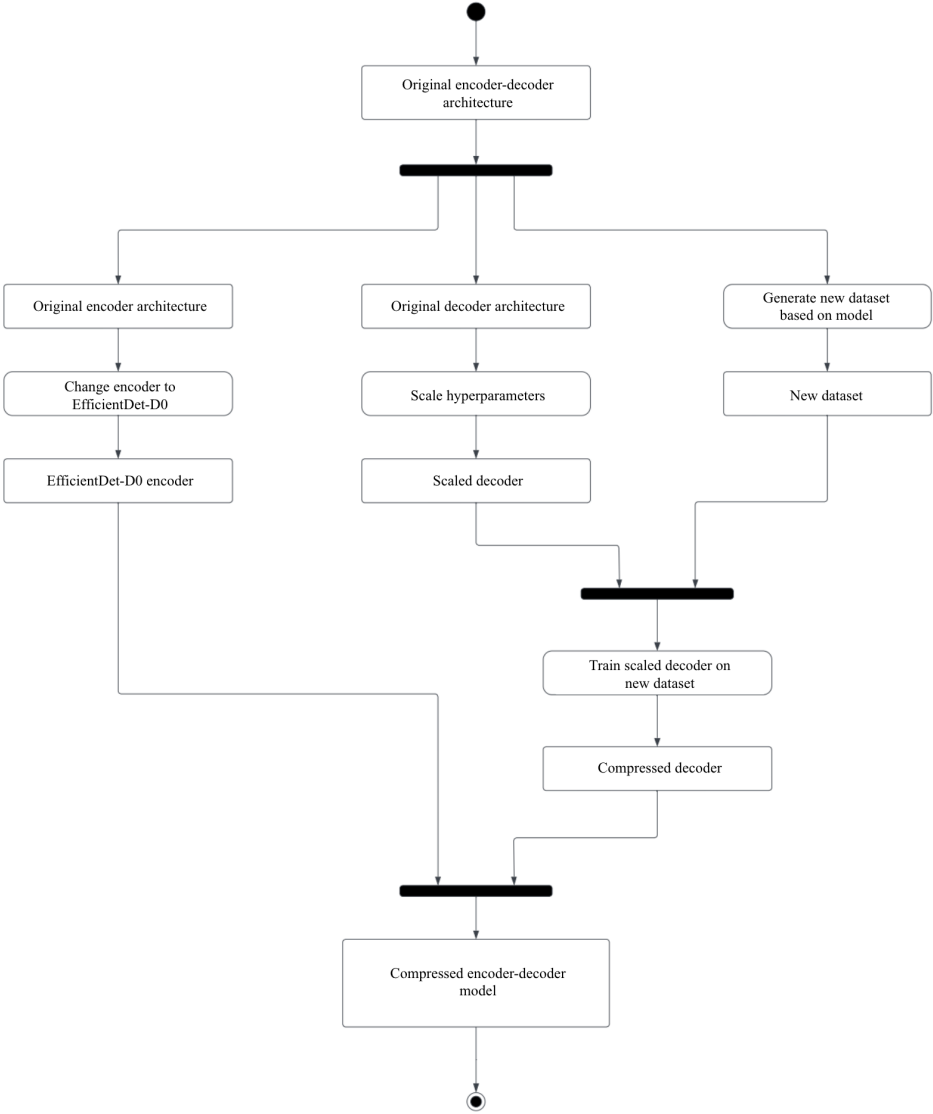


Fig. 2.6. Graphical representation of the second proposed compression method

architectural change, smaller decoder, and the best knowledge distillation method: EfficientDet-D0 encoder, decoder with a scale factor 0.25, knowledge distillation with beam sizes 10 and 3 for Up-Down and AoANet models, correspondingly. Also, a 95%-confidence interval was calculated for the proposed methods using the cross-validation algorithm on the same dataset with the fold number equal to 10.

Table 2.13. Evaluation results of the main metrics of all of the compression methods. The “Size” column refers to the whole model. C corresponds to CIDEr, and S corresponds to SPICE

Model	C	S	Size
Up-Down (Anderson et al., 2018)	120.1	21.4	661.4 MB
Up-Down compressed with the proposed first method	118.4 ± 1.2	20.8 ± 0.1	58.5 ± 13.8 MB
Up-Down compressed with the proposed second method	116.6 ± 0.9	20.5 ± 0.3	22.3 ± 5.1 MB
AoANet (Huang et al., 2019a)	129.8	22.4	791.8 MB
AoANet compressed with the proposed first method	127.4 ± 1.6	21.4 ± 0.4	34.8 ± 9.2 MB
AoANet compressed with the proposed second method	124.5 ± 1.3	21.5 ± 0.3	24.3 ± 5.6 MB
DistillVLM (Liu et al., 2020a)	131.7	23.5	154.5 MB
MiniVLM (Fang et al., 2021)	120.8	22.1	154.5 MB
LightCap (Wang et al., 2022)	136.6	24.2	145 MB

Table 2.14. Evaluation results of the other metrics of all of the compression methods. B@4 corresponds to BLEU-4, M corresponds to METEOR, and R corresponds to ROUGE-L

Model	B@4	M	R
Up-Down (Anderson et al., 2018)	36.3	27.7	56.9
Up-Down compressed with the proposed first method	35.3 ± 0.4	26.9 ± 0.3	56.3 ± 0.6
Up-Down compressed with the proposed second method	35.4 ± 0.5	26.9 ± 0.2	56.3 ± 0.6
AoANet (Huang et al., 2019a)	38.9	29.2	58.8
AoANet compressed with the proposed first method	38.3 ± 0.4	29.0 ± 0.7	58.3 ± 0.9
AoANet compressed with the proposed second method	37.5 ± 0.6	28 ± 0.5	57.5 ± 0.8
DistillVLM (Liu et al., 2020a)	39.2	29.7	-
MiniVLM (Fang et al., 2021)	35.6	28.7	-
LightCap (Wang et al., 2022)	40.1	29.9	-

As demonstrated, “First method” is the best according to the ratio between the compression effect and the metrics change. Compared to other methods, these provide better quality metrics than MiniVLM but lower quality metrics than DistilVLM and LightCap. However, all of them have model sizes around 150 MB, while this final model below 40 MB, which is more than a 70% reduction in size.

2.5. Conclusions of the Second Chapter

1. Three new encoders were proposed for use in image captioning: Faster R-CNN with MobileNetV3 backbone, EfficientDet-D0, and EfficientDet-D1. The use of smaller encoders designed for resource-constrained devices was shown to reduce the encoder size by 96.8%, from 468.4 MB to 15.1 MB, while only slightly decreasing the CIDEr and SPICE scores. The EfficientDet-D0 encoder was found to be the most suitable for the task.
2. Methods for reducing the size of the decoder showed good results with a scale factor of 0.5, resulting in models with comparable performance to the original models but with approximately three times less memory and parameters.
3. Two methods of pruning, l_1 pruning, and random pruning, were applied to the Up-Down and AoANet models. Overall, pruning was found to be able to reduce the model size by 10% while only losing 0.7% of the CIDEr score for the Up-Down model and reduce the model size by 50% while only losing 1.2% of the CIDEr score for the AoANet model.
4. It was shown that using dynamic quantization can decrease the size of a model by up to 50% with minimal loss of quality.
5. Sequence-level knowledge distillation is an efficient alternative to pruning and quantization techniques, especially for smaller models with a decoder scale factor of 0.25 and less powerful architectures like Up-Down. Using knowledge distillation can increase the CIDEr score by 2.2% from 114.3 to 116.8, and the SPICE score by 4% from 19.8 to 20.6.
6. Compared to other authors’ methods, these methods exceeded some of them in all metrics (both quality and size) and provided a much smaller size than all of them.

Increasing Image Captioning Neural Network Models' Quality

The following chapter of this dissertation focuses on improving the performance of image captioning models without making significant changes to their underlying architectures. The proposed approach is focused on several methods including encoder modifications, the use of pre-trained word embeddings, and the application of data augmentation techniques to the training dataset. The efficacy of these strategies will be evaluated on a range of image captioning models, selected based on their suitability for demonstrating the effects of the proposed methods.

The pursuit of improving image captioning models without making significant changes to their architectures is motivated by several considerations. First, it allows for a more efficient and cost-effective approach to improving the performance of existing models. Modifying the architecture of a model can be a time-consuming and resource-intensive process, whereas implementing strategies such as data augmentation can often be done relatively quickly and with minimal additional resources. Additionally, improving the quality of image captioning models without making significant changes to their architectures can help to advance the field of image captioning as a whole. By demonstrating the possibility of improving the performance of existing models without making significant changes to their architectures, the dissertation research aims to inspire others to explore similar strategies and potentially discover even more effective ways of improving the performance of image captioning models.

This chapter begins by examining the effect of changing the encoder in the Show, Attend and Tell model, one of the simplest image captioning architectures.

Specifically, the VGG encoder, typically used in this model, was replaced with a ResNet encoder to isolate the effect of this change from other modifications proposed in the literature. The conducted experiments showed that this change alone could lead to an improvement of 3.7% in relevant metrics.

Next, the use of pre-trained word embeddings was explored instead of training them from scratch within the context of the Show, Attend and Tell model. The conducted experiments demonstrated that this approach leads to an increase of 2.3 points in CIDEr and 0.6 in SPICE. Furthermore, it was found that GloVe embeddings were particularly well-suited to the image captioning task.

Subsequently, the use of data augmentation techniques was studied to augment the captions in the training dataset. Two new methods were proposed for this purpose: using synonyms and leveraging contextual word embeddings extracted from models such as BERT Devlin et al. (2019). The investigation focused on a range of configurations with varying amounts of dataset extensions and different levels of augmentation strength. The conducted experiments on three state-of-the-art image captioning models (namely, M^2 Transformers, AoANet, and X-LAN) showed that augmenting the initial dataset can improve CIDEr by 2.7 points and SPICE by 0.2 points without modifying the architecture of the initial model (and, therefore, not increasing its size).

The chapter concludes by examining the effect of the proposed augmentation techniques on already compressed models, building upon the work of the Second Chapter. The conducted experiments show that the quality of compressed models can be further improved by up to 1.6 points for CIDEr and 0.3 in SPICE relative to the best models trained using knowledge distillation.

The results of the experiments were presented at the conference and published in the proceedings (Atliha and Šešok, 2020a, 2021, 2022c) and published in the international journal (Atliha and Šešok, 2020b).

3.1. Experiment Setup

For all of the experiments in this chapter, the following evaluation setup was used.

MSCOCO by Lin et al. (2014), which is the largest and most used dataset for image captioning dataset was used for training, validating, and testing. Its standard version consists of 82,783 training images and 40,504 validation images. There are five different captions for each of the images. The standard Karpathy split by Karpathy and Fei-Fei (2015), which is used by most of the articles for result comparison, was used for offline evaluation. As a result, the final dataset consisted of 113,287 images for training, 5,000 images for validation, and 5,000 images for testing.

Also postprocessing was performed by replacing all the words that occurred less than five times in the final dataset with a special token <UNK>. Furthermore, because most captions were no more than 16 words in length, words were truncated to maintain the maximal length of the caption equal to 16.

Widely used metrics such as BLEU, METEOR, ROUGE-L, CIDEr, and SPICE were used to compare the efficiency of the methods, as well as metrics specific to the model compression task, such as memory consumption, number of parameters, and number of non-zero parameters.

A server with Intel(R) Xeon(R) CPU E5-2696 v4 @ 2.20GHz processor with 32 CPU cores, 120 GB RAM, Tesla V100 GPU card. Ubuntu 16.04 operation system was used as a computational device, PyTorch, and nlpaug libraries were installed.

3.2. Encoder Choice for Image Captioning Models

Although the previous chapter showed that the EfficientDet-D0 encoder is very efficient while this research focuses on memory consumption reduction, VGG and ResNet still remain of two the most popular image encoders for improving the quality of image captioning models. Thus, this sub-chapter focused on their effect from the quality perspective.

For this purpose, Show, Attend and Tell architecture (Fig. 3.1) was implemented using PyTorch.

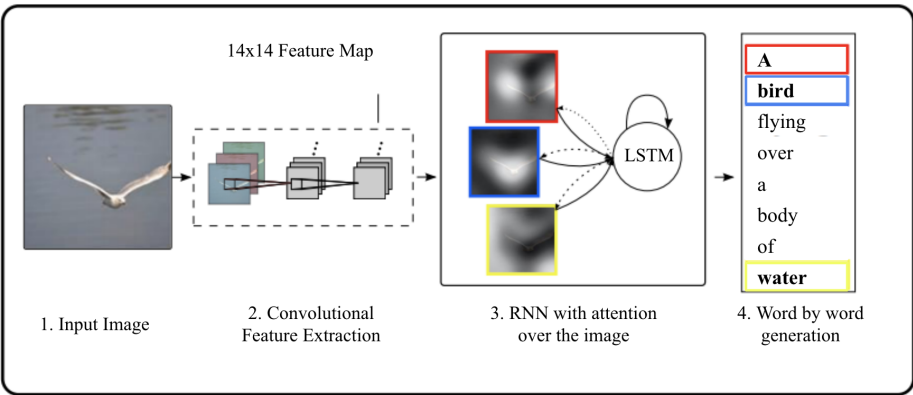


Fig. 3.1. Show, Attend and Tell architecture overview by Xu et al. (2015)

Built-in PyTorch pre-trained VGG19 and ResNet101 without fine-tuning were compared as encoders as they are the most popular architectures used as encoders

for image captioning tasks.

During the training, both models demonstrated similar performance in terms of accuracy and log-loss, although the model with the ResNet encoder exhibited slightly better results. The model with the VGG encoder achieved an accuracy of 0.765, while the model with the ResNet encoder reached 0.771 (Fig. 3.2). The models achieved log-losses of 3.12 and 3.08, respectively.

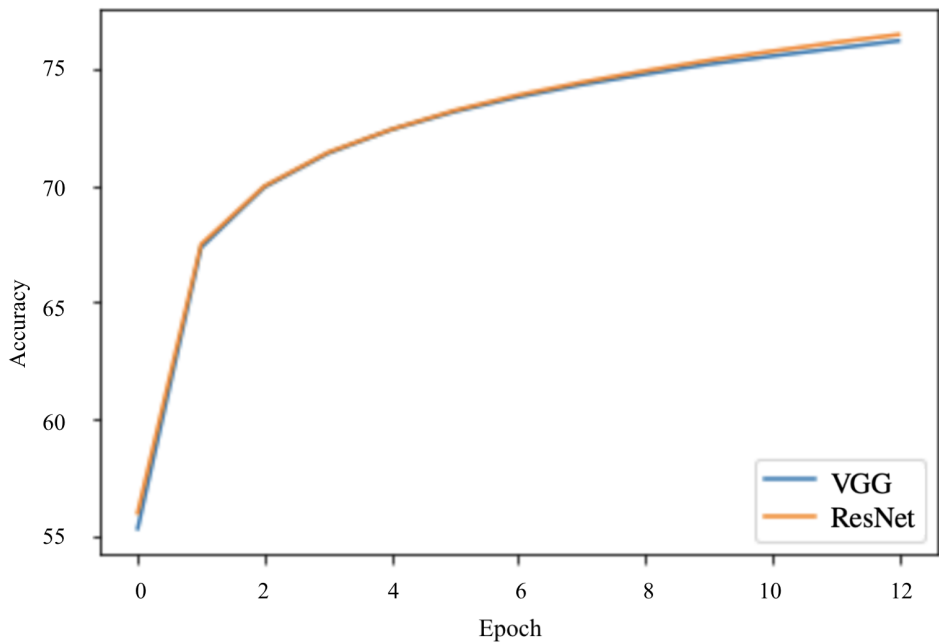


Fig. 3.2. Top-5 accuracy metric on the training dataset for models with different encoders

However, the models showed more distinct results during the validation phase. The model with the ResNet encoder not only outperformed the model with a VGG encoder in terms of directly minimized log-loss but also in terms of accuracy (Fig. 3.3). More importantly, the model with a ResNet encoder exhibited superior performance on BLEU-4, a metric specific to image captioning (Fig. 3.4). The difference in BLEU-4 between the two models was 0.008, which is quite significant. Additionally, the model with the ResNet encoder demonstrated faster training speed compared to the model with the VGG encoder. The former model reached 0.21 in BLEU-4 in the 5th epoch, while the latter model only achieved this performance in the 7th epoch.

Summarized results both for training and validation are given in Table 3.1.

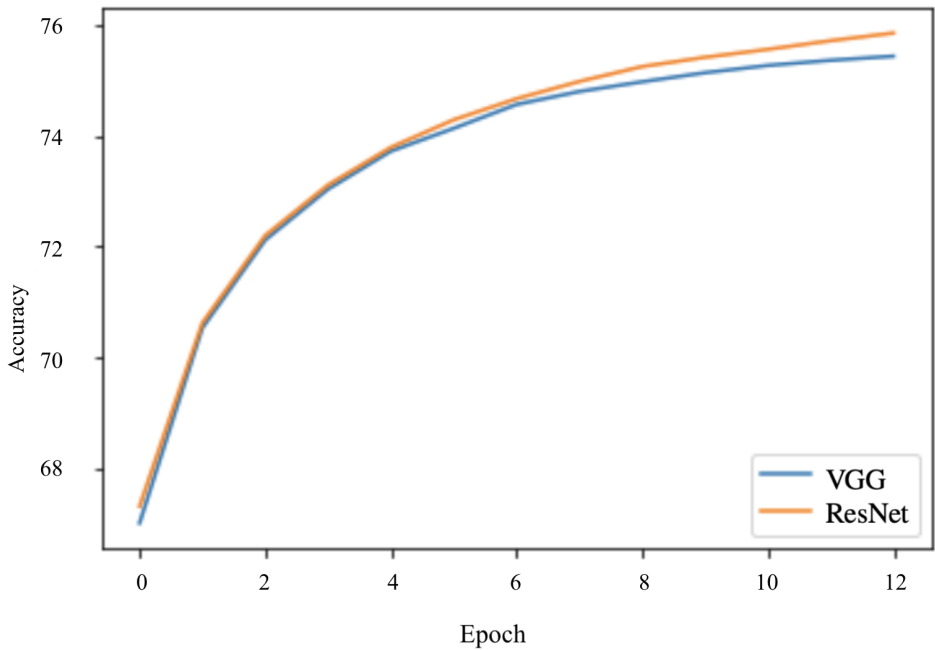


Fig. 3.3. Top-5 accuracy metric on the validation dataset for models with different encoders

Table 3.1. Models comparison

Phase	Encoder	Top-5 accuracy	Log-loss	BLEU-4
Training	VGG	0.762	3.146	-
Training	ResNet	0.765	3.129	-
Validation	VGG	0.754	3.216	21.8
Validation	ResNet	0.759	3.189	22.6

Based on the results presented above, ResNet outperforms VGG as an encoder for the image captioning task. This may be due to the fact that ResNet is generally considered a more powerful architecture for addressing many problems in image processing. The skip connections in its architecture allow the model to learn more efficiently, and its greater depth compared to VGG enables it to capture more complex concepts, making it a more effective encoder.

It is worth noting that being more powerful, ResNet is also a much bigger and heavier architecture than VGG. Thus, it makes it impossible to use it on memory-restricted devices. However, even VGG was too big (around 300 MB) for this.

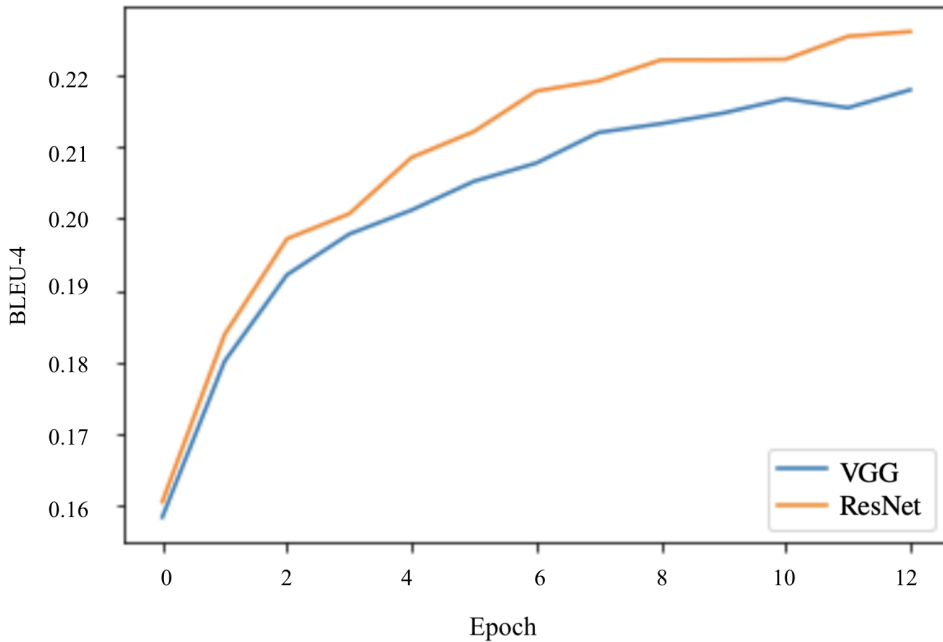


Fig. 3.4. BLEU-4 metric on the validation dataset for models with different encoders

3.3. Pre-Trained Word Embeddings

All modern methods for natural language processing tasks rely on vector representations of words. These can be either representations that are learned specifically for a given task or pre-trained vector representations that are learned from a large corpus of text data. Image captioning is no exception. While pre-trained vector representations are typically not used, they are instead trained along with the rest of the model during the training process for generating text descriptions of images. In this work, we explore whether the use of pre-trained vector representations for words can improve the quality of the image captioning model, as it has for other tasks.

The use of pre-trained word embeddings allows the model to represent words at a higher level of abstraction rather than simply as sequences of characters. There are several ways to establish a correspondence between words and vectors of numbers, ranging from simple one-hot encoding to more advanced methods such as Word2vec Mikolov et al. (2013) and GloVe Pennington et al. (2014). This paper compares different approaches to using pre-trained word embeddings to improve the training of image captioning models.

3.3.1. One-hot Encoding

One-hot encoding is a method of representing words as vectors of numbers. Given a dictionary of fixed length n where each word is assigned a unique number m , one-hot encoding represents a word with a vector of size n where all elements are zero except for the element at index m , which is one. For example, if the word “cat” is assigned the number 2 in the dictionary, then its one-hot encoded vector would be $[0, 0, 1, 0, \dots, 0]$.

One-hot encoding is simple and effective at representing the difference between words, but it has several drawbacks. First, the vector representations for words can be very large and sparse, making them inefficient for use in some applications. Additionally, one-hot encoding does not capture the semantic relationships between words, making it unsuitable for tasks such as image captioning. For these reasons, other methods of representing words as vectors may be more appropriate in some situations. For instance, instead of using one-hot encoding, one could use word embeddings, where each word is represented by a low-dimensional dense vector that captures semantic relationships between words. This can be achieved using such techniques as Word2vec or GloVe.

3.3.2. Word-to-Vector Embeddings

Word2vec is a popular method for obtaining vector representations of words that capture the semantic similarity between words. This is achieved by training a vector representation for each word so that it is easy to reconstruct the vector representations of words in their context using the word's vector, or vice versa. There are two main algorithms for training word vectors with word2vec: Skip-gram and Continuous Bag of Words (CBOW). When trained on a large corpus of text, the resulting word vectors are highly versatile and can be used in many natural language processing tasks.

To train the word vectors, word2vec uses a shallow neural network with a single hidden layer. Given a sequence of words $w_1, w_2, \dots, w_n$, the input to the network is the one-hot encoded vectors for each word, and the output is a predicted probability distribution over the words in the vocabulary. The objective of the training is to maximize the likelihood of the observed words given the predicted probabilities. By optimizing the weights of the network, the word vectors are learned so that they capture the semantic relationships between words. The exact details of the training process depend on the specific algorithm being used.

The two algorithms in word2vec, Skip-gram and Continuous Bag of Words (CBOW), differ in how they train the neural network. In Skip-gram, the input is the center word, and the output is the context words, whereas in CBOW, the input is the context words, and the output is the center word. Mathematically, Skip-gram can be

expressed as:

$$L(\theta) = \frac{1}{T} \sum_{t=1}^T \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j}|w_t; \theta), \quad (3.1)$$

where w_t is the center word, w_{t+j} is the context words, and $p(w_{t+j}|w_t)$ is the predicted probability of the context word w_{t+j} given the center word w_t . The objective is to maximize the likelihood of the observed context words given the predicted probabilities.

CBOW can be expressed similarly, with the input and output reversed:

$$L(\theta) = \frac{1}{T} \sum_{t=1}^T \sum_{-m \leq j \leq m, j \neq 0} \log p(w_t|w_{t+j}; \theta), \quad (3.2)$$

where w_t is the center word and w_{t+j} are the context words.

In both algorithms, the predicted probabilities are computed using the softmax function, which takes as input the dot product of the word vectors and a weight matrix:

$$p(w_i|w_j) = \frac{\exp(\mathbf{v}_i^T \mathbf{W} \mathbf{v}_j)}{\sum_k \exp(\mathbf{v}_k^T \mathbf{W} \mathbf{v}_j)}. \quad (3.3)$$

Here, $p(w_i|w_j)$ represents the predicted probability of the word w_i given the context of the word w_j . The numerator of this expression is the dot product of the word vectors for words i and j , which has been multiplied by the weight matrix W . The denominator is a normalization factor that ensures the probabilities sum to 1.

3.3.3. Global Vectors for Word Representation

GloVe by Pennington et al. (2014) is a method for learning vector representations of words, also known as word embeddings. Unlike word2vec, which uses shallow neural networks and requires large amounts of labeled training data, GloVe is based on unsupervised learning algorithms and can be trained on a large corpus of text data without the need for labels.

At the core of the GloVe method is the idea of using the co-occurrence statistics of words in a corpus of text to learn their vector representations. In other words, the model counts how often each word appears in the same context as every other word in the corpus. This co-occurrence information is then used to train a model that learns to predict the co-occurrences of words from their vector representations.

Once the model is trained, the resulting word vectors can be used in a similar way to word2vec vectors for such tasks as natural language processing and text

analysis. The GloVe method has been shown to produce high-quality word vectors that capture the semantic and syntactic relationships between words in a corpus.

The mathematical details of the GloVe method are somewhat complex, but at a high level, the method involves the following steps:

First, the co-occurrence statistics of all word pairs in the corpus are collected and stored in a word-word co-occurrence matrix X , where the element $X_{i,j}$ represents the number of times the word i appears in the same context as the word j in the corpus.

Next, the matrix is “de-meaning” by subtracting the row and column means from each element, giving the new matrix Y where each element is defined as:

$$Y_{i,j} = X_{i,j} - \frac{\sum_{k=1}^V X_{i,k}}{V} - \frac{\sum_{k=1}^V X_{k,j}}{V} + \frac{\sum_{k=1}^V \sum_{l=1}^V X_{k,l}}{V^2}. \quad (3.4)$$

The matrix Y is then factorized into the product of two matrices, U and V , where each row of U and V corresponds to the vector representation of a word in the vocabulary. This factorization can be done using a variety of techniques, such as singular value decomposition (SVD) or alternating least squares (ALS).

The resulting matrices U and V can then be used to obtain the word vectors for each word in the vocabulary. For example, the vector representation of word i can be obtained by taking the i th row of either U or V .

3.3.4. Empirical Evaluation and Discussion of Results of the Word Embeddings Choice for Image Captioning Models

To start, the performance of the developed model was compared with embeddings trained from scratch for the problem, as well as two versions of pretrained Word2vec and GloVe embeddings. The plot of the metric values on the validation dataset per training epoch can be seen in Figs. 3.5 and 3.6.

As can be seen in the figures, models with pre-trained embeddings tend to perform well from the very first training epoch. In contrast, the model without pre-trained embeddings only achieves a 54.1 CIDEr score and a 12.1 SPICE score after the first epoch. In comparison, the models with word2vec and GloVe pre-trained embeddings have a 58.4 and a 59.7 CIDEr score and a 12.4 and a 12.4 SPICE score, respectively. This can be explained by the fact that pre-trained embeddings already contain a significant amount of information about the semantics of words in the language, allowing the model to take advantage of this knowledge from the beginning. However, over time, the difference between models with pre-trained and trained-from-scratch embeddings tends to become smaller, although the model with pre-trained embeddings still performs better overall. For example, after the last epoch, the model without pre-trained embeddings achieves a 76.8 CIDEr score

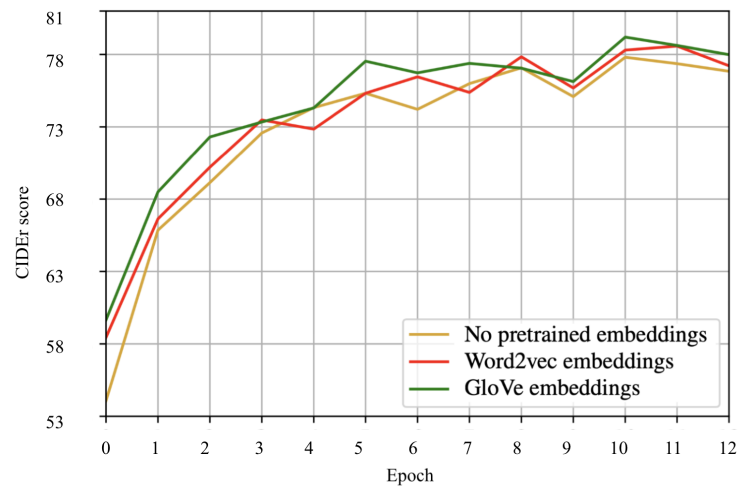


Fig. 3.5. Validation CIDEr score per epochs during a training phase for the model without pre-trained embeddings and for the models with Word2vec and GloVe pretrained embeddings

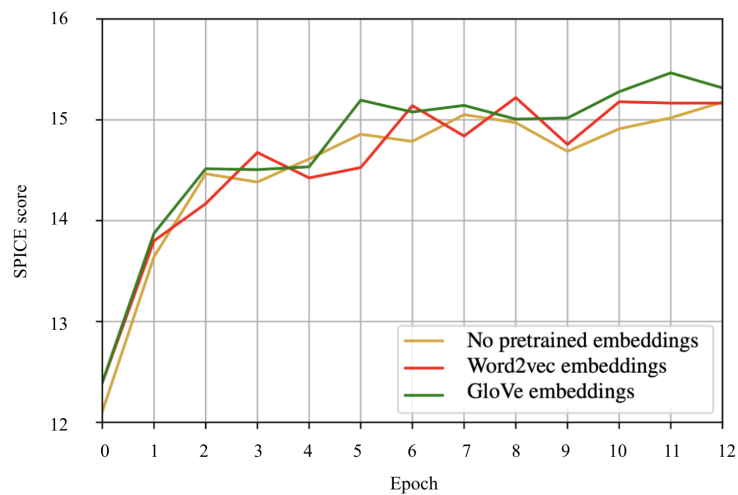


Fig. 3.6. Validation SPICE score per epochs during a training phase for the model without pre-trained embeddings and for the models with Word2vec and GloVe pretrained embeddings

and a 15.2 SPICE score, while the models with pre-trained embeddings achieve a 77.2 and a 78 CIDEr score and a 15.2 and a 15.3 SPICE score, respectively, for word2vec and GloVe embeddings. The difference is only about 0.4 to 1.2 points for the CIDEr score and about 0 to 0.1 points for the SPICE score.

It is also worth noting that the model with GloVe tends to perform better than the model with word2vec. The best model with word2vec pre-trained embeddings reaches a 78.6 CIDEr score and a 15.2 SPICE score, while the best model with GloVe pre-trained embeddings reaches a 79.2 CIDEr score and a 15.5 SPICE score. This suggests that using GloVe pre-trained embeddings may be a more effective approach for this particular problem.

The next experiment aimed to investigate the effects of fine-tuning pre-trained embeddings during overall model training on a specific dataset. As the GloVe embeddings performed the best in the previous experiment, the validation metrics were reported only for these embeddings. The metric values on the validation dataset during training can be seen in Figs. 3.7 and 3.8.

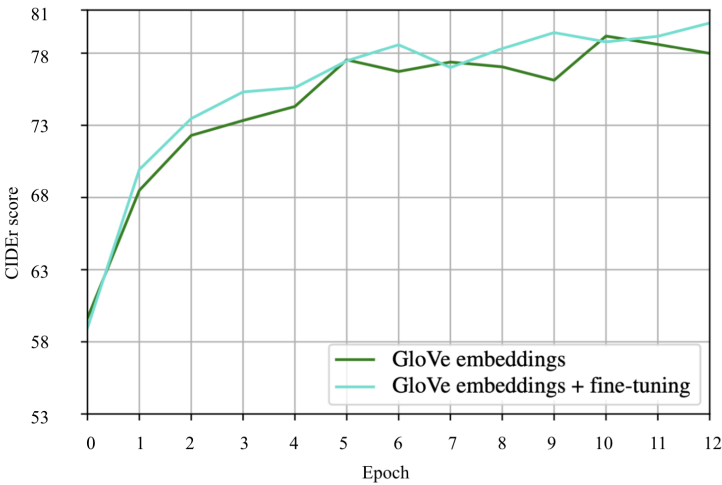


Fig. 3.7. Validation CIDEr scores per epochs during a training phase for models using pre-trained GloVe embeddings with and without fine-tuning

Based on the experiment results fine-tuning the word embeddings certainly helps to improve the quality of the model. The CIDEr and SPICE scores increased from 79.2 and 15.5 to 79.4 and 15.7, respectively, when the word embeddings were fine-tuned. This is likely due to the fact that the image captioning problem has its own specific semantics, which may require a slightly different vector representation of words to achieve better results. Fine-tuning the word embeddings allows the

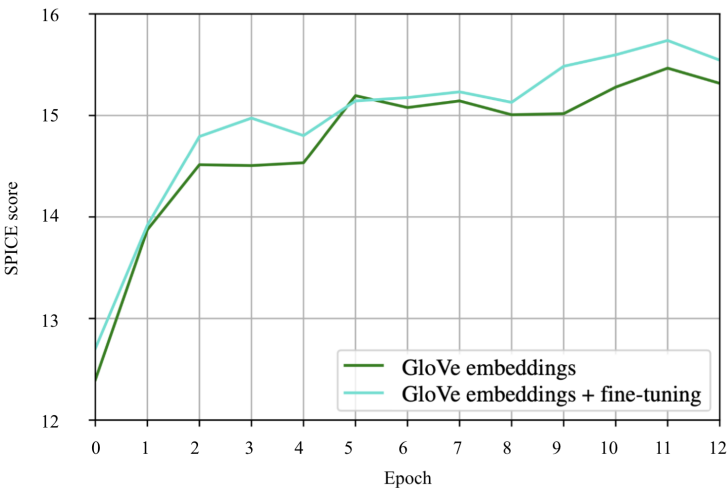


Fig. 3.8. Validation SPICE scores per epochs during a training phase for models using pre-trained GloVe embeddings with and without fine-tuning

model to adjust the vectors to better fit the specific semantics of the image captioning problem, leading to improved performance.

The final results of comparing all of the models on all of the metrics on the test dataset are presented in Tables 3.2 and 3.3. These tables show that the best option is to use pre-trained GloVe embeddings that are fine-tuned during the training of the entire model. This is consistent with how word embeddings are typically used in other natural language processing tasks. Using fine-tuned GloVe embeddings allows the model to take advantage of the knowledge and information contained in the pre-trained vectors while also allowing the vectors to be adjusted to better fit the specific semantics of the image captioning problem. This leads to improved performance on the test dataset.

Table 3.2. Models' main metrics on a test dataset

	ROUGE-L	CIDEr	SPICE
No pre-trained	50	77.8	15.2
Word2vec	50	78.4	15.2
Word2vec + fine-tuning	50.1	78.6	15.3
GloVe	50.1	79.2	15.7
GloVe + fine-tuning	50.3	80.1	15.8

According to the tables, using pre-trained GloVe embeddings helps to improve the CIDEr score of the model by 1.4 points and the SPICE score by 0.5 points,

Table 3.3. Models' other metrics on a test dataset

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
No pretrained	68.1	51	36.9	26.4	21.9
Word2vec	68.2	51.2	37.2	26.6	22.2
Word2vec +fine-tuning	68.7	51.4	37.3	26.7	22.3
GloVe	68.7	51.6	37.5	26.8	22.3
GloVe + fine-tuning	69	51.6	37.6	26.9	22.4

compared to a model without pre-trained embeddings. Additionally, further fine-tuning of the GloVe embeddings leads to an increase in the CIDEr score by 0.9 points and in the SPICE score by 0.1 points. This demonstrates the effectiveness of using pre-trained GloVe embeddings, as well as the benefits of fine-tuning these embeddings to better fit the specific semantics of the image captioning problem. Overall, this approach leads to improved performance on the test dataset.

3.4. Captions Augmentation Methods for Dataset Extension

Image captioning models, despite their high quality, have a limited ability to generate a variety of descriptions for objects in different situations. This limitation is largely due to the fact that current training datasets, while large in terms of the number of images, have a relatively small number of captions per image. To address this issue, the training dataset was expanded using text augmentation methods, including the use of synonyms as a baseline and the application of the state-of-the-art language model BERT for calculating contextualized word embeddings. These methods allowed for increasing the diversity of the training data, improving the ability of the image captioning model to generate a wider range of descriptions.

3.4.1. Synonymous Augmentation for Dataset Extension

Unlike image and speech processing, where adding random noise to the input signal can be a useful augmentation technique, this approach is not suitable for text augmentation. The relative order of letters and their presence in a word can significantly affect the semantic meaning of the word itself, so simply adding random noise to the input text is not likely to be effective. The best method for text augmentation is to rephrase sentences as a person would, but this approach is complicated to implement and scale to large training datasets. the proposed method is simpler but still effective and is synonymous replacement augmentation, first introduced for natural language processing tasks by Zhang et al. (2015).

Let I be an image from a training set, and let $C = c_1, \dots, c_k$ be a set of captions corresponding to that image, where each caption is a sequence of words $c_i = (w_{i,1}, w_{i,2}, \dots, w_{i,l_i})$, and l_i is the length of the i -th caption. We also fix a synonymous thesaurus T , and let $T(w_{i,j}) = (s_{i,j,1}, s_{i,j,2}, \dots, s_{i,j,m_{i,j}})$ be the list of synonyms for the word $w_{i,j}$ sorted in descending order of semantic closeness to the most frequently seen meaning of the word $w_{i,j}$, where $m_{i,j}$ is the number of synonyms of the word $w_{i,j}$ in the thesaurus T .

To generate a new caption c'_i based on an existing caption c_i , the following procedure is performed. First, fix a probability p . Then, for every word $w_{i,j} \in c_i$ that has synonyms in a dictionary (i.e., for which $m_{i,j} > 1$), replace it with one of its synonyms with probability p . To determine which synonym to use, fix another probability q . With probability q , replace the word with the most semantically similar synonym $s_{i,j,1}$. If the word is not replaced by the first synonym, replace it with the second most similar synonym $s_{i,j,2}$ with probability q , and so on. This means that the probability of replacing a word with synonym $s_{i,j,r}$ is equal to q^r , and it exponentially decreases with the semantic similarity of the synonym to the original word. This operation of replacing a word with one of its synonyms is performed independently for each word in a sentence.

After generating a new caption c'_i based on an existing caption c_i , this operation is repeated d times, where d is called the augmentation coefficient. This means that if an image has k captions, applying augmentation with coefficient d will result in kd captions, effectively increasing the size of the training set by a factor of d .

A graphical representation of the synonyms augmentation method can be found in Fig. 3.9.

3.4.2. Contextualized Word Embedding Augmentation for Dataset Extension

The contextualized word embedding approach, as described by Fadaee et al. (2017) and Kobayashi (2018), can be used to augment a given set of sentences that describe an image. Specifically, given an image I and a set of sentences $C = c_1, \dots, c_k$ describing that image, each sentence is a sequence of words $c_i = (w_{i,1}, w_{i,2}, \dots, w_{i,l_i})$.

To augment these sentences, a language model LM can be used that can predict the probability that a particular word w will occur in a certain context. Given a caption c_i and the j -th word of that caption, the context of that word is the entire caption except for the word itself, i.e. $c_i \setminus w_{i,j} = (w_{i,1}, w_{i,2}, \dots, w_{i,j-1}, w_{i,j+1}, \dots, w_{i,l_i})$. Therefore, $LM(c_i, j) = P(\cdot | c_i \setminus w_{i,j})$ is a probability distribution over the words that can occur at position j in caption c_i , considering the context of that word.

To generate new, augmented sentences based on a given caption c_i , the following procedure can be applied. First, fix a probability p that each word in the caption

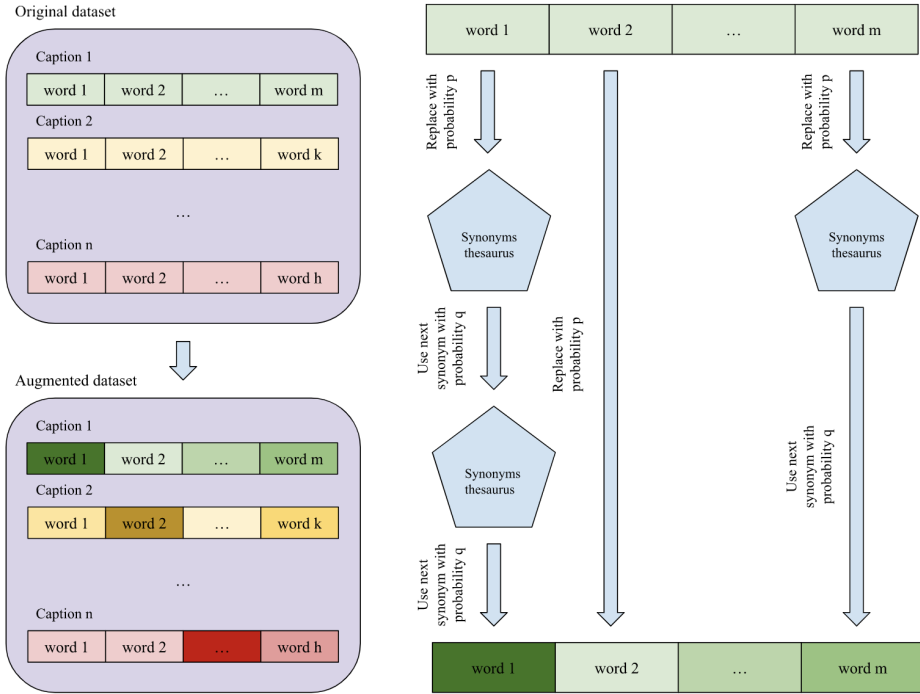


Fig. 3.9. Graphical representation of the synonyms augmentation method

should be replaced with another word. Then, for each word $w_{i,j}$ in the caption, calculate $LM(c_i, j)$. From this probability distribution, generate a new word $w'_{i,j} \sim LM(c_i, j)$ and use it to replace $w_{i,j}$ in the original caption, resulting in a new augmented sentence c'_i .

By repeating this process for each word in the caption, and repeating the entire process for each of the k captions in the set C , kd augmented sentences can be generated describing the image I , where d is the number of times the augmentation was performed. This allows us for increasing the diversity and richness of the set of sentences that describe the image, potentially improving the performance of downstream natural language processing tasks.

- To augment a caption, a language model LM can be used that can predict the probability that a particular word w will occur in a certain context,
- Given a caption c_i and the j -th word of that caption, the context of that word is the entire caption except for the word itself, i.e. $c_i \setminus w_{i,j} = (w_{i,1}, w_{i,2}, \dots, w_{i,j-1}, w_{i,j+1}, \dots, w_{i,l_i})$
- Therefore, $LM(c_i, j) = P(\cdot | c_i \setminus w_{i,j})$ is a probability distribution over the

words that can occur at position j in caption c_i , taking into account the context of that word.

The use of contextualized word embedding models, such as BERT by Devlin et al. (2019), DistilBERT by Sanh et al. (2019), RoBERTA by Liu et al. (2019c), or XLNet by Yang et al. (2019c), is particularly well-suited for the augmentation procedure described above. These models are designed to capture the context in which a word appears, allowing them to better predict the probability that a particular word will occur in a given context. This is especially important for the augmentation procedure, as we are trying to generate new words that are likely to occur in the given context, while maintaining the overall meaning of the sentence.

In comparison to other language models, contextualized word embedding models have been shown to perform well on a wide range of natural language processing tasks, and have achieved state-of-the-art performance on many benchmarks. This makes them an attractive choice for use in the augmentation procedure, as they are likely to produce high-quality augmented sentences that accurately capture the context of the original sentences. Additionally, these models are typically trained on large amounts of data, which allows them to capture rich and nuanced information about the relationships between words, further improving their ability to generate appropriate augmented sentences.

For our dissertation's studies the BERT model by Devlin et al. (2019) was used as a model for contextualized word embeddings. BERT, or Bidirectional Encoder Representations from Transformers, is a state-of-the-art language model that is trained to predict missing words in a sentence based on their context. To do this, the model is first trained on a large corpus of text, during which some of the words in each sentence are replaced with a special token ([MASK]). The model is then trained to predict these missing words based on the context of the surrounding words.

In addition to predicting missing words, BERT is also trained to understand the relationship between sentences. This is done by training the model to predict whether one sentence is a logical continuation of another. As a result, BERT is pre-trained in an unsupervised manner on a large corpus of text, and is able to generate high-quality vector representations of words that capture their context and meaning. All of these features make BERT well-suited for use in the augmentation procedure described above.

A graphical representation of the contextualized word embeddings augmentation method can be found in Fig. 3.10.

3.4.3. Empirical Evaluation of Augmentation Techniques

The undertaken studies compared five different augmentation techniques for augmenting a dataset of image captions. Augmentation was only used for the training

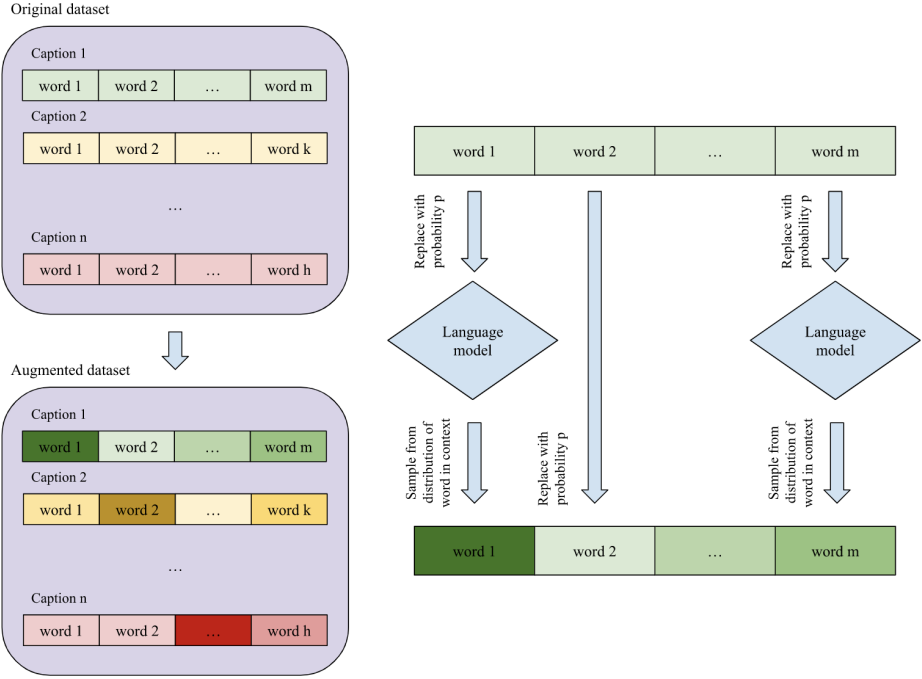


Fig. 3.10. Graphical representation of the contextualized word embeddings augmentation method

part of the dataset, and a model was used without any augmentation as a baseline for comparison. In other methods, the dataset was augmented using BERT with augmentation factors d equal to 2 and 3, as well as synonyms with an augmentation factor of 2. In all cases the default value of the replacement rate p was set to 0.1, except during the studies on the effect of the replacement rate.

Models were trained for 19 epochs, first 12 of them were with cross-entropy loss and then the rest were trained using the self-critical approach. This change could explain huge spikes in metrics after the 12th epoch.

To evaluate the effectiveness of these augmentation techniques, a state-of-the-art model by Cornia et al. (2020b) was trained on the augmented datasets. Extensive experiments were conducted to determine the best augmentation method for this model. The results were also confirmed on other state-of-the-art models, e.g., by Huang et al. (2019a) and Pan et al. (2020), using the best variant of the dataset chosen based on the previous experiments with Cornia et al. (2020b). All of the models were trained using open source code released by the authors of the corresponding

papers. The `nlpaug` library was used for augmentation. It is important to note that, when trained using the open-source code, all of the used models showed slightly worse performance than reported in the corresponding papers.

3.4.4. Results and Discussion of Augmentation Techniques

Fig. 3.11 and 3.12 show the effect of using BERT with $d = 2$ for different values of the replacement rate p on augmentation performance. It appears that when p is set to 0.5, the model performs worse compared to when p is set to 0.1. This may be due to the fact that a large replacement rate results in a significant loss of information from the original caption, which leads to less grammatically correct and human-like sentences. In contrast, a lower replacement rate of 0.1 is able to add a variation to the dataset without damaging the meaning and grammatical structure of the sentences, resulting in improved performance.

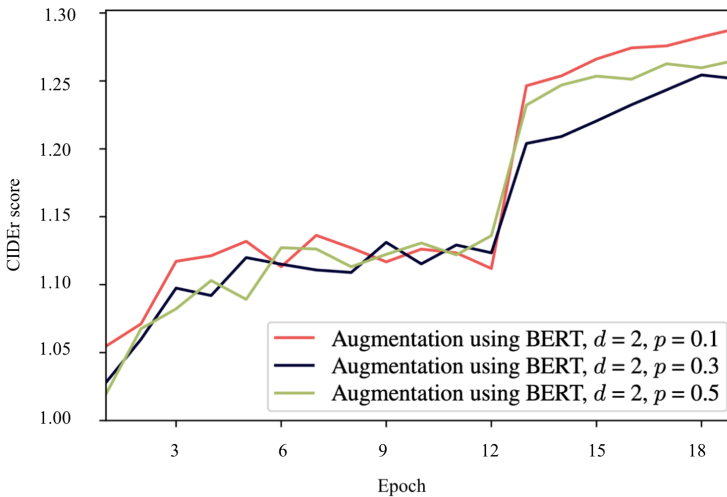


Fig. 3.11. Validation CIDEr scores per epoch on datasets with BERT augmentation with $d=1$ and $p=0.1, 0.3, 0.6$

Figs. 3.13 and 3.14 allow for comparing the performance of models trained using various augmentation techniques - synonyms and BERT - with a model trained on the original dataset. The model trained on a dataset augmented with synonyms exhibits slightly better performance than the original model. Furthermore, the model trained on a dataset augmented using BERT shows significantly better performance compared to both the original model and the model trained on the synonymous augmented dataset. This suggests that augmenting the dataset with BERT is an

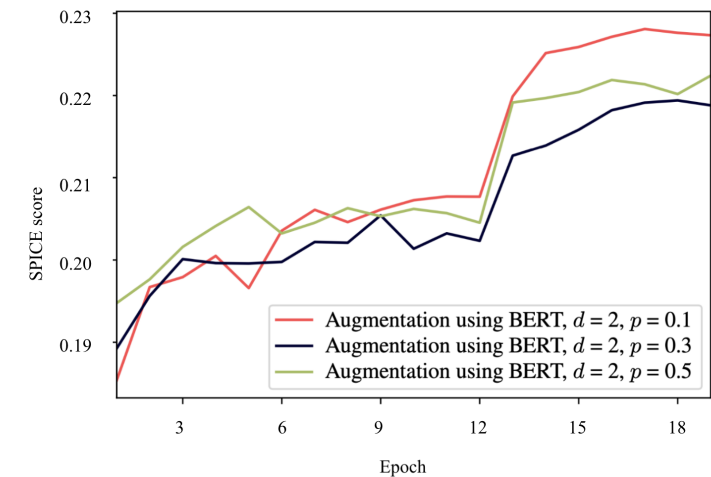


Fig. 3.12. Validation SPICE scores per epoch on datasets with BERT augmentation with $d=1$ and $p=0.1, 0.3, 0.6$

effective method for improving model performance.

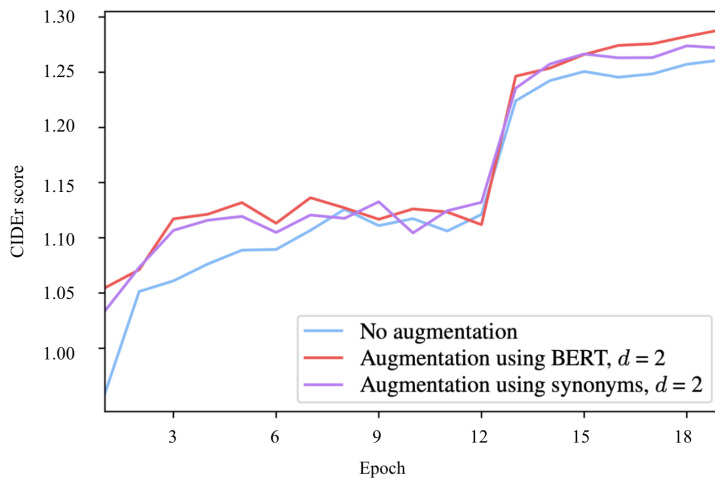


Fig. 3.13. Validation CIDEr scores per epoch on dataset without augmentation, on a dataset with BERT augmentation with $d=2$, and on a dataset with synonyms augmentation with $d=2$

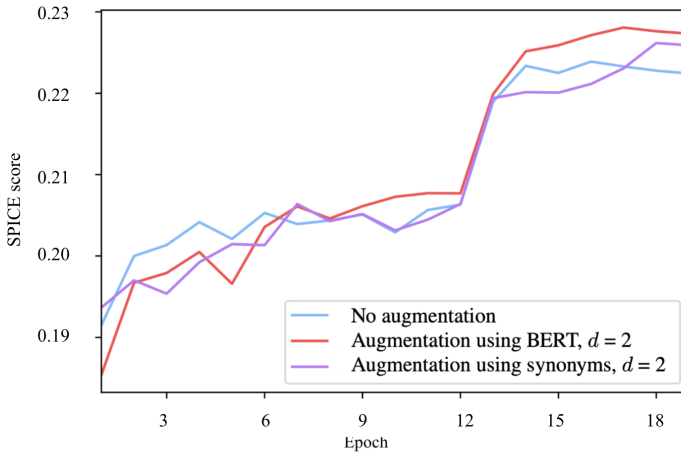


Fig. 3.14. Validation SPICE scores per epoch on dataset without augmentation, on a dataset with BERT augmentation with $d=2$, and on a dataset with synonyms augmentation with $d=2$

In Figures 3.15 and 3.16, we compare the performance of models trained on BERT augmented datasets with $d=2$ and $d=3$ and $p=0.1$ (which has been shown to be the most promising replacement rate). The comparison reveals that training on a more augmented dataset does not result in improved performance. In fact, training on a dataset that has been augmented three times results in worse performance compared to training on a dataset augmented two times. This suggests that there are limits to the effectiveness of this augmentation method and that a significant increase in dataset size beyond two times does not improve the quality of the model.

The final test scores for all trained Cornia et al. (2020b) models are summarized in Table 3.4. The model trained on the 2-times increased dataset obtained using BERT with $p = 0.1$ augmentation shows the best results across almost all metrics, significantly outperforming the model trained on the original dataset by 2.7 points for CIDEr and 0.2 points for SPICE. This provides evidence for the effectiveness of the proposed augmentation method in improving the quality of image captioning models. Augmentation can be used as a simple and effective way to improve the performance of existing state-of-the-art approaches without making any modifications to the underlying model.

Table S3.2 and Table 3.6 summarize the results for all three models trained with BERT augmentation with $d=2$ and $p=0.1$. Also, a 95%-confidence interval was calculated for the proposed methods using a cross-validation algorithm on the same dataset with a fold number equal to 10. The results show that all models trained with

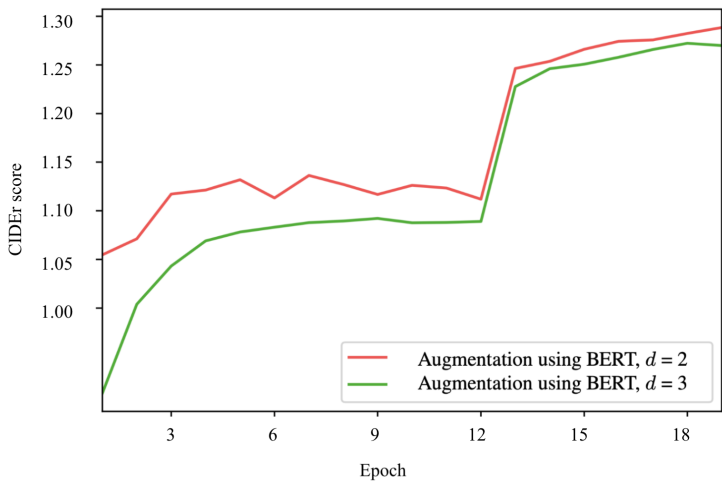


Fig. 3.15. Validation CIDEr scores per epoch on a dataset with BERT augmentation with $d=2$ and $d=5$

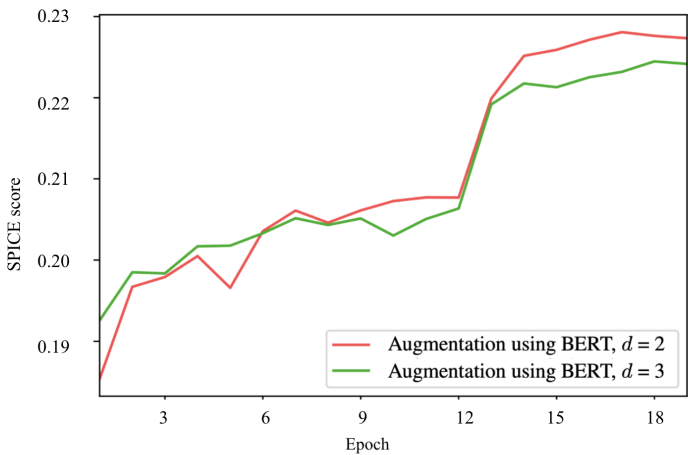


Fig. 3.16. Validation SPICE scores per epoch on a dataset with BERT augmentation with $d=2$ and $d=5$

Table 3.4. Evaluation results of the Cornia et al. (2020b) model trained with proposed augmentation methods

Augmentation Type	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
No augmentation	38.3	28.6	58.0	126.1	22.6
BERT, $d = 2$, $p = 0.1$	37.8	28.9	58.3	128.8	22.8
BERT, $d = 2$, $p = 0.3$	37.3	28.5	57.8	125.4	21.9
BERT, $d = 2$, $p = 0.5$	37.3	28.8	58.0	126.5	22.3
BERT, $d = 3$	37.9	28.6	57.9	127.2	22.4
Synonyms, $d = 2$	37.7	28.7	57.8	127.4	22.2

augmentation outperform the corresponding models trained without augmentation. This supports the conclusions about the benefits of using the proposed augmentation method for training state-of-the-art image captioning models. This suggests that augmenting the training data with BERT can be an effective way to improve model performance.

Table 3.5. Evaluation results of main metrics of all three tested models trained with the dissertation's augmentation methods

Model	CIDEr	SPICE
M^2 Transformers (Cornia et al., 2020b), no augmentation	126.1	22.6
M^2 Transformers, augmentation	128.8 ± 1.4	22.8 ± 0.3
AoANet (Huang et al., 2019a), no augmentation	125.9	22.3
AoANet, augmentation	128.4 ± 1.9	22.5 ± 0.1
X-LAN (Pan et al., 2020), no augmentation	126.1	22.8
X-LAN, augmentation	128.6 ± 1.1	22.9 ± 0.1

Table 3.6. Evaluation results of other metrics of all three tested models trained with the dissertation's augmentation methods

Model	BLEU-4	METEOR	ROUGE-L
M^2 Transformers, no augmentation	38.3	28.6	58.0
M^2 Transformers, augmentation	37.8 ± 0.4	28.9 ± 0.2	58.3 ± 0.1
AoANet, no augmentation	38.2	28.6	57.9
AoANet, augmentation	37.6 ± 0.5	28.8 ± 0.3	58.0 ± 0.1
X-LAN, no augmentation	38.8	28.9	58.2
X-LAN, augmentation	37.9 ± 0.1	29.2 ± 0.1	58.3 ± 0.0

To the best of our knowledge, it is the first time that text augmentation techniques were applied for the extension of the image captioning dataset, which makes it impossible to compare with the other methods of the same nature from different authors.

3.4.5. Quantitative Analysis of Augmentation Techniques

Several examples were selected for captions generated by the resulting models on a test dataset. These examples are shown in Figs. 3.17 and 3.18. In these figures, “Ground truth” denotes a real human-generated caption from the test set used for evaluating model performance, and “Original” denotes a caption generated by a model trained on the original dataset without augmentation. The examples demonstrate that augmenting the training dataset with BERT helps the resulting model generate more elegant and rich sentences compared to a model trained on the original dataset without augmentation.



Original: a fire hydrant on the side of a street

BERT, $d = 2$, $p = 0.1$: a black and silver fire hydrant on the side of a street

BERT, $d = 2$, $p = 0.3$: a fire hydrant on the side of a street with orange cones

BERT, $d = 2$, $p = 0.5$: a fire hydrant on a side of a street with two cones

Synonyms, $d = 2$: a fire hydrant on the side of a street a cones

Ground truth: a bicycle is lying on the sidewalk beside a fire hydrant

Fig. 3.17. Examples of captions generated by the proposed models. Image source: MS COCO dataset

Additionally, some examples of augmenting the original captions using synonyms and BERT are presented in Figs. 3.19 and 3.20. These examples illustrate that both augmentation methods are able to diversify the captions, providing the model with additional opportunities to learn more complex and general concepts about the textual description of images. However, since the augmentation does not take the content of the image into account, sometimes the augmented captions do not accurately reflect the contents of the image. This simulates the type of noise that may be present in descriptions created by humans. Overall, the augmented captions are similar to the ground truth captions, although they are not perfect. This



Original: a small airplane is parked on a runway
 BERT, $d = 2$, $p = 0.1$: the front of a propeller airplane on a runway street
 BERT, $d = 2$, $p = 0.3$: a small plane is sitting on the runway
 BERT, $d = 2$, $p = 0.5$: an airplane is sitting on the runway at the airport
 Synonyms, $d = 2$: a small plane is parked on the runway cones
 Ground truth: a small airplane taking off from an airport runway

Fig. 3.18. Examples of captions generated by the proposed models. Image source: MS COCO dataset

suggests that augmentation can be a useful tool for improving the performance of image captioning models.

3.5. Caption Augmentations for Compressed Models

As described in the Second Chapter, the augmentation techniques discussed in Sub-chapter 3.3 can be applied to compressed models to further improve their performance. Since these augmentation techniques do not affect the size of the model, they can be combined with compression techniques to create high-quality models that are relatively small in terms of memory consumption. This can be particularly useful for applications where memory constraints are a concern. By using both compression and augmentation, it is possible to achieve excellent performance without sacrificing the model size.

To further improve the performance of compressed models, it is proposed to combine augmentation methods with sequence-level knowledge distillation. These two techniques are similar in the sense that both involve generating new captions for images, which can be used to create a new training dataset. By combining augmentation and knowledge distillation, it is possible to create a larger training dataset that can be used to train a high-quality image captioning model.



Original: A restaurant has modern wooden tables and chairs

BERT: Luxurious restaurant with modern wooden tables and chairs

BERT: A cafe has modern wooden walls and chairs

Synonyms: A dining room has modern wooden form and chairs

Synonyms: A restoration has modern wooden tables and directors

Fig. 3.19. Examples of proposed captions augmentation. Image source: MS COCO dataset

3.5.1. Augmentation with Knowledge Distillation

The proposed method is based on the captions augmentation introduced in Sub-chapter 3.3 and the knowledge distillation approach for image captioning models described in Sub-chapter 2.4. These two techniques were combined in the following way.

Firstly, we use a large "teacher" model M_t trained on the original image captioning dataset to generate new captions for the images in the dataset. The teacher model takes an image as input and produces five human-like captions describing the image. This creates a new dataset that can be used as the input for text augmentation methods.

Formally, let the original dataset be

$$D_{initial} = \{(I_a, \{c_{a,1}, c_{a,2}, \dots, c_{a,5}\}), a \in [1, n]\}, \quad (3.5)$$

where I_a is the image and $c_{a,1}, c_{a,2}, \dots, c_{a,5}$ are the corresponding captions. Using



Original: A large bus and some people on the street

BERT: One city bus and fifteen people on the street

BERT: Another large bus carrying some people crowded the street

Synonyms: A enormous bus involved some people requests the street

Synonyms: A widespread minibus and some people on the highways

Fig. 3.20. Examples of proposed captions augmentation. Image source: MS COCO dataset

the teacher model M_t , a new dataset is generated

$$D_{distillation} = \{(I_a, M_t(I_a)), a \in [1, n]\}, \quad (3.6)$$

where $M_t(I_a) = c'a, 1, c'a, 2, \dots, c'a, 5$ are the captions generated by the teacher model.

Next, the dataset $D_{distillation}$ is augmented using one of the methods described in 3.4. Let A be the augmentation method, which takes a caption as input and returns an augmented variant of that caption. The augmented dataset is then given by

$$D_{augmented} = \{(I_a, \{A(c'_{a,1}), \dots, A(c'_{a,5})\}), a \in [1, n]\} \quad (3.7)$$

This augmented dataset is used as the final dataset for training image captioning models, effectively combining the knowledge distillation and text augmentation methods. This allows the model to benefit from the improved diversity and quality of the augmented captions, potentially leading to better performance on the image captioning task.

3.5.2. Empirical Evaluation of Augmentation with Knowledge Distillation

The proposed comparison of methods uses two different model architectures, Up-Down and AoANet, to provide thorough verification of the results. In accordance with the Second Chapter, model variations were used with an EfficientDet-d0 encoder and a particular decoder size, specifically the version with a decoder scaling coefficient $\gamma = 0.25$. It is worth noting that this model size is equal to only 7.2 MB for the Up-Down model and 9.2 MB for the AoANet model.

It was decided to focus on $\gamma = 0.25$ models rather than $\gamma = 0.5$ models, as the former are of less model size while still having lower initial quality compared to the latter. Additionally, augmentation methods were applied to the models obtained using the best knowledge distillation methods to obtain the best possible quality with the small model size.

For augmentation, a replacement rate of $p=0.1$ and an augmentation factor of $d=2$ were used, as these were the best parameters according to the experiments described earlier. Also, the use of synonyms and BERT augmentation techniques were compared for the developed models, as both showed promising results. For all experiments, the AoANet model was used with a beam size of 10 trained on the initial dataset as the “teacher” model, as it has been shown to provide the best results in the corresponding sub-chapter.

Model implementation was used from the publicly available <https://github.com/ruotianluo/ImageCaptioning.pytorch> (accessed on 2 April 2022) repository. They were implemented using the PyTorch framework. The public repository <https://github.com/zylo117/Yet-Another-EfficientDet-Pytorch> (accessed on 2 April 2022) was used to obtain EfficientDet-d0 model. Models were trained for 30 epochs in a standard way and then for 10 more epochs in a self-critical way replicating Rennie et al. (2017).

For text augmentation method implementation, the public repository <https://github.com/makcedward/nlpaug> (accessed on 2 April 2022) was used.

3.5.3. Results and Discussion of Augmentation with Knowledge Distillation

The results of a comparison of different techniques for the Up-Down model are provided in Table 3.7. The “Original” entry refers to a model without any size compression techniques applied. The “With KD” entry represents the model with the best performance from Sub-chapter 2.4 after applying knowledge distillation. The “With synonyms” and “With BERT” entries refer to models that have been augmented using synonyms and BERT, respectively.

As can be seen from the table, both synonym augmentation and the use of BERT

Table 3.7. Evaluation results for Up-Down Model

	B@4	METEOR	ROUGE-L	CIDEr	SPICE
Original	33.2	26.2	55.5	114.3	19.8
With KD	35.4	26.9	56.3	116.6	20.5
With synonyms	35.2	27	56.1	117.6	20.7
With BERT	35.2	26.8	56.2	118.2	20.8

have been shown to improve the quality of the model. Synonyms augmentation increased the CIDEr score by 1 point and the SPICE score by 0.2 points, while the use of BERT word embeddings increased both metrics even further, by 1.6 points for CIDEr and 0.3 points for SPICE. This aligns with our previous research, which indicates that BERT is a more effective augmentation approach for the image captioning task than using synonyms.

However, as can also be seen from the table, the effect of knowledge distillation is greater than that of augmentation. Specifically, knowledge distillation resulted in an increase of 2.3 points in the CIDEr score and 0.7 points in the SPICE score, compared to an increase of 1.6 points in the CIDEr score and 0.3 points in the SPICE score for augmentation. This may be because augmentation only slightly changes the training data, resulting in potentially noisy data, while knowledge distillation is based on the use of high-quality models with known performance.

The results of a comparison of similar experiments for the AoANet model are shown in Table 3.8. These data confirm the conclusions drawn from the results in Table 3.7, namely that BERT augmentation provides the best results, increasing the CIDEr score by 1 point (compared to a 0.3 point increase for synonyms augmentation) and the SPICE score by 0.2 points (compared to a 0.1 point increase for synonyms augmentation). Furthermore, the performance of BERT augmentation is less affected than knowledge distillation, which increased the CIDEr score by 1.6 points and the SPICE score by 0.6 points.

Table 3.8. Evaluation results for AoANet Model

	B@4	METEOR	ROUGE-L	CIDEr	SPICE
Original	36.4	28.6	57.4	122.9	20.9
With KD	37.5	28	57.5	124.5	21.5
With synonyms	38	28.6	57.2	124.8	21.6
With BERT	38.1	28.8	57.3	125.5	21.7

It is worth noting that augmentation appears to have a greater effect on the Up-Down model than on AoANet. As AoANet is initially a better-performing model than Up-Down, this may suggest that the effect of augmentation decreases as the quality of the initial model increases. This could be because high-quality models may already capture most of the relevant information in the training data.

3.6. Conclusions of the Third Chapter

1. The image captioning model with a ResNet encoder outperforms the model with a VGG encoder in terms of directly minimized log-loss, accuracy, and a BLEU-4. The model with a ResNet encoder also demonstrated faster training speed.
2. The method was proposed for using pre-trained word embeddings such as word2vec and GloVe in image captioning models. Models with pre-trained word embeddings and fine-tuning them during the training tend to perform generally better, especially from the very first training epoch. The best model uses GloVe embeddings and increases the CIDEr score by 3% and the SPICE score by 4%.
3. The use of text augmentation methods was proposed, including the use of synonyms and contextualized word embeddings. A smaller replacement rate (0.1) of the original caption with BERT-generated sentences results in improved performance, while a larger replacement rate (0.5) leads to worse performance. Augmenting the dataset with BERT also improves model performance compared to augmenting with synonyms or using the original dataset. The best method helps to increase the CIDEr score by 2.4% and the SPICE score by 0.9%.
4. An algorithm combining sequence-level knowledge distillation method with captions augmentations was proposed to improve compressed image captioning models. The results show that both synonyms augmentation and the use of BERT word embeddings for an augmenting dataset for knowledge distillation improve the quality of the model, with BERT showing the most improvement (1.4% for CIDEr and 1.4% for SPICE).

General Conclusions

1. The literature review showed an immense interest in the image captioning topic among researchers. The typical image captioning model has an encoder-decoder architecture, where the encoder is typically a vision model (such as VGG or ResNet), and the decoder is typically a language model (such as RNN or LSTM). To improve the quality of these models, different architectures (including attention-based and transformer-based) are often used without considering their impact on memory consumption. In the majority of papers, the same datasets (such as MS COCO and Flickr30k) are used with the same preprocessing techniques. However, the question of extending these datasets remains open.
2. Novel methods proposed for image captioning model compression allow reducing the model size by up to 91.2%, losing only 1.4% in the CIDEr score and 2.8% in the SPICE score. The best overall method is to use the EfficientDet-D0 encoder, decoder with a scale factor of 0.5, pruning with a coefficient dependent on the model architecture and use quantization. Such a method allows for achieving quality metrics comparable with the other author's methods being much less memory consuming.
3. The use of text augmentation methods was proposed, including the use of synonyms and contextualized word embeddings. A smaller replacement rate (0.1) of the original caption with BERT-generated sentences results in improved performance, while a larger replacement rate (0.5) leads to worse

performance. Augmenting the dataset with BERT also improves model performance compared to augmenting with synonyms or using the original dataset. There are limits to the effectiveness of this augmentation method, and training on a dataset that has been augmented three times results in worse performance compared to a dataset augmented two times. The model trained on a 2-times increased dataset using BERT with a 0.1 replacement rate outperforms the model trained on the original dataset by 3.4% for CIDEr and 5% for SPICE.

References

- Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., & Anderson, P. (2019). Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 8948–8957).
- Aker, A., & Gaizauskas, R. (2010). Generating image descriptions using dependency relational patterns. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1250–1258).
- Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6077–6086).
- Anwar, S., Hwang, K., & Sung, W. (2017). Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3): 1–18.
- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.
- Bai, S., & An, S. (2018). A survey on automatic image caption generation. *Neurocomputing*, 311: 291–304.
- Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., & Plank, B. (2016). Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 55: 409–442.
- Chatterjee, M., & Schwing, A. G. (2018). Diverse and coherent paragraph generation

- from images. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 729–744).
- Chen, C., Mu, S., Xiao, W., Ye, Z., Wu, L., & Ju, Q. (2019). Improving image captioning with conditional generative adversarial nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33 (pp. 8142–8150).
- Chen, L., Jiang, Z., Xiao, J., & Liu, W. (2021). Human-like controllable image captioning with verb-specific semantic roles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16846–16856).
- Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., & Chua, T.-S. (2017). Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5659–5667).
- Chen, S., & Zhao, Q. (2018). Boosted attention: Leveraging human attention for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 68–84).
- Chen, X., Ma, L., Jiang, W., Yao, J., & Liu, W. (2018). Regularizing rnns for caption generation by reconstructing the past with the present. In *Proceedings of the IEEE Conference on computer vision and pattern recognition* (pp. 7995–8003).
- Choi, J., Wang, Z., Venkataramani, S., Chuang, P. I.-J., Srinivasan, V., & Gopalakrishnan, K. (2018). Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*.
- Chunseong Park, C., Kim, B., & Kim, G. (2017). Attend to you: Personalized image captioning with context sequence memory networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 895–903).
- Cornia, M., Baraldi, L., & Cucchiara, R. (2019). Show, control and tell: A framework for generating controllable and grounded captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8307–8316).
- Cornia, M., Baraldi, L., & Cucchiara, R. (2020a). Smart: training shallow memory-aware transformers for robotic explainability. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1128–1134). IEEE.
- Cornia, M., Baraldi, L., Serra, G., & Cucchiara, R. (2018). Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(2): 1–21.
- Cornia, M., Stefanini, M., Baraldi, L., & Cucchiara, R. (2020b). Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10578–10587).
- Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

- Cui, Y., Yang, G., Veit, A., Huang, X., & Belongie, S. (2018). Learning to evaluate image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5804–5812).
- Deshpande, A., Aneja, J., Wang, L., Schwing, A. G., & Forsyth, D. (2019). Fast, diverse and accurate image captioning guided by part-of-speech. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10695–10704).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, (pp. 4171–4186).
- Elliott, D., & Keller, F. (2013). Image description using visual dependency representations. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1292–1302).
- Fadaee, M., Bisazza, A., & Monz, C. (2017). Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 567–573).
- Fang, Z., Wang, J., Hu, X., Liang, L., Gan, Z., Wang, L., Yang, Y., & Liu, Z. (2022). Injecting semantic concepts into end-to-end image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18009–18019).
- Fang, Z., Wang, J., Hu, X., Wang, L., Yang, Y., & Liu, Z. (2021). Compressing visual-linguistic model via knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1428–1438).
- Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. In *European conference on computer vision* (pp. 15–29). Springer.
- Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., & Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, Vol. 26. Curran Associates, Inc.
- Fu, K., Jin, J., Cui, R., Sha, F., & Zhang, C. (2016). Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts. *IEEE transactions on pattern analysis and machine intelligence*, 39(12): 2321–2334.
- Gan, C., Gan, Z., He, X., Gao, J., & Deng, L. (2017). Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3137–3146).
- Gao, J., Wang, S., Wang, S., Ma, S., & Gao, W. (2019). Self-critical n-step training for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6300–6308).
- Ge, H., Yan, Z., Zhang, K., Zhao, M., & Sun, L. (2019). Exploring overall contextual information for image captioning in human-like cognitive style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1754–1763).

- Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. *Low-Power Computer Vision*, (pp. 291–326).
- Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).
- Gong, Y., Wang, L., Hodosh, M., Hockenmaier, J., & Lazebnik, S. (2014). Improving image-sentence embeddings using large weakly annotated photo collections. In *European conference on computer vision* (pp. 529–545). Springer.
- Gu, J., Cai, J., Wang, G., & Chen, T. (2018). Stack-captioning: Coarse-to-fine learning for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- Guo, L., Liu, J., Tang, J., Li, J., Luo, W., & Lu, H. (2019). Aligning linguistic words and visual semantic units for image captioning. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 765–773).
- Guo, L., Liu, J., Zhu, X., Yao, P., Lu, S., & Lu, H. (2020). Normalized and geometry-aware self-attention network for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10327–10336).
- Gupta, A., Verma, Y., & Jawahar, C. (2012). Choosing linguistics over vision to describe images. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 26 (pp. 606–612).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- He, S., Liao, W., Tavakoli, H. R., Yang, M., Rosenhahn, B., & Pugeault, N. (2020a). Image captioning through image transformer. In *Proceedings of the Asian Conference on Computer Vision* (pp. 153–169). University of Glasgow.
- He, Y., Ding, Y., Liu, P., Zhu, L., Zhang, H., & Yang, Y. (2020b). Learning filter pruning criteria for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2009–2018).
- He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., & Han, S. (2018). Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 784–800).
- Hendricks, L. A., Venugopalan, S., Rohrbach, M., Mooney, R., Saenko, K., & Darrell, T. (2016). Deep compositional captioning: Describing novel object categories without paired training data. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–10).
- Herdade, S., Kappeler, A., Boakye, K., & Soares, J. (2019). Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32.

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8): 1735–1780.
- Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., & Peste, A. (2021). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *The Journal of Machine Learning Research*, 22(1): 10882–11005.
- Hossain, M. Z., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6): 1–36.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019). In searching for mobilenetv3. (pp. 1314–1324).
- Hu, X., Gan, Z., Wang, J., Yang, Z., Liu, Z., Lu, Y., & Wang, L. (2021). Scaling up vision-language pre-training for image captioning. *arXiv preprint arXiv:2111.12233*.
- Huang, L., Wang, W., Chen, J., & Wei, X.-Y. (2019a). Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4634–4643).
- Huang, L., Wang, W., Xia, Y., & Chen, J. (2019b). Adaptively aligned image captioning via adaptive attention time. *Advances in Neural Information Processing Systems*, (pp. 8940–8949).
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. (pp. 2704–2713).
- Janisha, T., & Purushotham, S. (2016). A survey of neural network algorithms used for image annotation. *IJOAB JOURNAL*, 7(5): 236–252.
- Ji, J., Luo, Y., Sun, X., Chen, F., Luo, G., Wu, Y., Gao, Y., & Ji, R. (2021). Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35 (pp. 1655–1663).
- Jia, X., Gavves, E., Fernando, B., & Tuytelaars, T. (2015). Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2407–2415).
- Jiang, W., Ma, L., Jiang, Y.-G., Liu, W., & Zhang, T. (2018). Recurrent fusion network for image captioning. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 499–515).
- Kafle, K., Yousefhusien, M., & Kanan, C. (2017). Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation* (pp. 198–202).
- Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3128–3137).

- Karpathy, A., Joulin, A., & Fei-Fei, L. (2014). Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2* (pp. 1889–1897).
- Ke, L., Pei, W., Li, R., Shen, X., & Tai, Y.-W. (2019). Reflective decoding network for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8888–8897).
- Kim, D.-J., Choi, J., Oh, T.-H., & Kweon, I. S. (2019). Dense relational captioning: Triple-stream networks for relationship-based captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6271–6280).
- Kiros, R., Salakhutdinov, R., & Zemel, R. (2014a). Multimodal neural language models. In *International conference on machine learning* (pp. 595–603). PMLR.
- Kiros, R., Salakhutdinov, R., & Zemel, R. S. (2014b). Unifying visual-semantic embeddings with multimodal neural language models.
- Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. (pp. 452–457).
- Krause, J., Johnson, J., Krishna, R., & Fei-Fei, L. (2017). A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 317–325).
- Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1): 32–73.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.
- Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., & Berg, T. L. (2011). Baby talk: Understanding and generating simple image descriptions. In *CVPR 2011* (pp. 1601–1608).
- Kumar, A., & Goel, S. (2017). A survey of evolution of image captioning techniques. *International Journal of Hybrid Intelligent Systems*, 14(3): 123–139.
- Kuznetsova, P., Ordonez, V., Berg, A., Berg, T., & Choi, Y. (2012). Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 359–368).
- Kuznetsova, P., Ordonez, V., Berg, T. L., & Choi, Y. (2014). Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, 2: 351–362.
- Lee, N., Ajanthan, T., & Torr, P. (2018). Snip: Single-shot network pruning based on connection sensitivity. In *Proceedings of the International Conference on Learning Representations* (pp. 291–326).

- Li, G., Zhu, L., Liu, P., & Yang, Y. (2019a). Entangled transformer for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 8928–8937).
- Li, J., Li, D., Xiong, C., & Hoi, S. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning* (pp. 12888–12900). PMLR.
- Li, R., Wang, Y., Liang, F., Qin, H., Yan, J., & Fan, R. (2019b). Fully quantized network for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2810–2819).
- Li, S., Kulkarni, G., Berg, T., Berg, A., & Choi, Y. (2011). Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning* (pp. 220–228).
- Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision* (pp. 121–137). Springer.
- Li, Y., Yao, T., Pan, Y., Chao, H., & Mei, T. (2019c). Pointing novel objects in image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 12497–12506).
- Liang, X., Hu, Z., Zhang, H., Gan, C., & Xing, E. P. (2017). Recurrent topic-transition gan for visual paragraph generation. In *Proceedings of the IEEE international conference on computer vision* (pp. 3362–3371).
- Lin, C.-Y., & Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (p. 605). Association for Computational Linguistics.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision* (pp. 740–755). Springer.
- Liu, F., Liu, Y., Ren, X., He, X., & Sun, X. (2019a). Aligning visual regions and textual concepts for semantic-grounded image representations. *Advances in Neural Information Processing Systems*, 32: 1015–1028.
- Liu, F., Ren, X., Liu, Y., Lei, K., & Sun, X. (2020a). Exploring and distilling cross-modal information for image captioning. *arXiv preprint arXiv:2002.12585*.
- Liu, F., Ren, X., Wu, X., Ge, S., Fan, W., Zou, Y., & Sun, X. (2020b). Prophet attention: Predicting attention with future attention. *Advances in Neural Information Processing Systems*, 33: 1865–1876.

- Liu, S., Zhu, Z., Ye, N., Guadarrama, S., & Murphy, K. (2017). Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision* (pp. 873–881).
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. (pp. 21–37).
- Liu, W., Chen, S., Guo, L., Zhu, X., & Liu, J. (2021). Cpnr: Full transformer network for image captioning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 19165–19174).
- Liu, X., Xu, Q., & Wang, N. (2019b). A survey on deep neural network-based image captioning. *The Visual Computer*, 35(3): 445–470.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019c). Roberta: A robustly optimized bert pretraining approach. *Advances in Neural Information Processing Systems*, (pp. 1751–1762).
- Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 375–383).
- Lu, J., Yang, J., Batra, D., & Parikh, D. (2018). Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7219–7228).
- Luo, Y., Huang, Z., Zhang, Z., Wang, Z., Li, J., & Yang, Y. (2019). Curiosity-driven reinforcement learning for diverse visual paragraph generation. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 2341–2350).
- Luo, Y., Ji, J., Sun, X., Cao, L., Wu, Y., Huang, F., Lin, C.-W., & Ji, R. (2021). Dual-level collaborative transformer for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35 (pp. 2286–2293).
- Mao, Y., Zhou, C., Wang, X., & Li, R. (2018). Show and tell more: Topic-oriented multi-sentence image captioning. In *IJCAI* (pp. 4258–4264).
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference for Learning Representations* (pp. 10578–10587).
- Mitchell, M., Dodge, J., Goyal, A., Yamaguchi, K., Stratos, K., Han, X., Mensch, A., Berg, A., Berg, T., & Daumé III, H. (2012). Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 747–756).
- Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2019). Pruning convolutional neural networks for resource efficient inference. In *Proceedings International Conference on Learning Representations*.
- OpenAI (2023). *Gpt-4 technical report*.

- Ordonez, V., Kulkarni, G., & Berg, T. (2011). Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems*, Vol. 24 (pp. 21–35).
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.
- Pan, J.-Y., Yang, H.-J., Duygulu, P., & Faloutsos, C. (2004). Automatic image captioning. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, Vol. 3 (pp. 1987–1990). IEEE.
- Pan, Y., Yao, T., Li, Y., & Mei, T. (2020). X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10971–10980).
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 311–318). Association for Computational Linguistics.
- Park, C. C., Kim, B., & Kim, G. (2018). Towards personalized image captioning via multi-modal memory networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(4).
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).
- Qin, Y., Du, J., Zhang, Y., & Lu, H. (2019). Look back and predict forward in image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8367–8375).
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748–8763). PMLR.
- Ramanishka, V., Das, A., Zhang, J., & Saenko, K. (2017). Top-down visual saliency guided by captions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7206–7215).
- Rampal, H., & Mohanty, A. (2020). Efficient cnn-lstm based image captioning using neural network compression. *arXiv e-prints*, (pp. arXiv–2012).
- Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2016). Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

- Ren, Z., Wang, X., Zhang, N., Lv, X., & Li, L.-J. (2017). Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 290–298).
- Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7008–7024).
- Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Proceedings of the Conference on Neural Information Processing Systems*.
- Shabir, S., & Arafat, S. Y. (2018). An image conveys a message: A brief survey on image description generation. In *2018 1st International Conference on Power, Energy and Smart Grid (ICPESG)* (pp. 1–6). IEEE.
- Sharma, H., Agrahari, M., Singh, S. K., Firoj, M., & Mishra, R. K. (2020). Image captioning: a comprehensive survey. In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)* (pp. 325–328). IEEE.
- Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., & Keutzer, K. (2021). How much can clip benefit vision-and-language tasks? In *Proceedings of the 10th International Conference on Learning Representations*.
- Shi, Z., Zhou, X., Qiu, X., & Zhu, X. (2020). Improving image captioning with better use of caption. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7454–7464).
- Shuster, K., Humeau, S., Hu, H., Bordes, A., & Weston, J. (2019). Engaging image captioning via personality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 12516–12526).
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*.
- Staniūtė, R., & Šešok, D. (2019). A systematic literature review on image captioning. *Applied Sciences*, 9(10): 2024.
- Stefanini, M., Cornia, M., Baraldi, L., Cascianelli, S., Fiameni, G., & Cucchiara, R. (2022). From show to tell: a survey on deep learning-based image captioning. IEEE.
- Sugano, Y., & Bulling, A. (2016). Seeing with humans: Gaze-assisted neural image captioning. *Computer Science and Psychology*.
- Sun, C., Gan, C., & Nevatia, R. (2015). Automatic concept discovery from parallel text and visual corpora. In *Proceedings of the IEEE international conference on computer vision* (pp. 2596–2604).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
- Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 5100–5111).
- Tan, J. H., Chan, C. S., & Chuah, J. H. (2022). End-to-end supermask pruning: Learning to prune image captioning models. *Pattern Recognition*, 122: 108366.
- Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105–6114). PMLR.
- Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781–10790).
- Tanaka, H., Kunin, D., Yamins, D. L., & Ganguli, S. (2020). Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in neural information processing systems*, 33.
- Tavakoli, H. R., Shetty, R., Borji, A., & Laaksonen, J. (2017). Paying attention to descriptions generated by image captioning models. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2487–2496).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4566–4575).
- Venugopalan, S., Anne Hendricks, L., Rohrbach, M., Mooney, R., Darrell, T., & Saenko, K. (2017). Captioning images with diverse objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5753–5761).
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. (pp. 3156–3164).
- Wang, C., Yang, H., Bartz, C., & Meinel, C. (2016). Image captioning with deep bidirectional lstms.
- Wang, C., Yang, H., & Meinel, C. (2018). Image captioning with deep bidirectional lstms and multi-task learning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(2s): 1–20.
- Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, (p. 11).

- Wang, L., Bai, Z., Zhang, Y., & Lu, H. (2020). Show, recall, and tell: Image captioning with recall mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34 (pp. 12176–12183).
- Wang, N., Xie, J., Luo, H., Cheng, Q., Wu, J., Jia, M., & Li, L. (2022). Efficient image captioning for edge devices. *arXiv preprint arXiv:2212.08985*.
- Wang, Y., & Cook, D. (2020). Brain: A bidirectional generative adversarial networks for image captions. In *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence* (pp. 1–6).
- Wang, Y., Lin, Z., Shen, X., Cohen, S., & Cottrell, G. W. (2017). Skeleton key: Image captioning by skeleton-attribute decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7272–7281).
- Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., & Cao, Y. (2021). Simvlm: Simple visual language model pretraining with weak supervision. In *Proceedings of the International Conference on Learning Representations*.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3).
- Wu, Y., Zhu, L., Jiang, L., & Yang, Y. (2018). Decoupled novel object captioner. In *Proceedings of the 26th ACM international conference on Multimedia* (pp. 1029–1037).
- Xiao, X., & Wang, Z. (2019). Autoprune: Automatic network pruning by regularizing auxiliary parameters. *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 32.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning* (pp. 2048–2057).
- Yan, S., Wu, F., Smith, J. S., Lu, W., & Zhang, B. (2018). Image captioning using adversarial networks and reinforcement learning. In *2018 24th International Conference on Pattern Recognition (ICPR)* (pp. 248–253). IEEE.
- Yang, X., Tang, K., Zhang, H., & Cai, J. (2019a). Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10685–10694).
- Yang, X., Zhang, H., & Cai, J. (2019b). Learning to collocate neural modules for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4250–4260).
- Yang, Y., Teo, C., Daumé III, H., & Aloimonos, Y. (2011). Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 444–454).
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019c). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems* (pp. 5754–5764).

- Yang, Z., Yuan, Y., Wu, Y., Cohen, W. W., & Salakhutdinov, R. R. (2016). Review networks for caption generation. *Advances in neural information processing systems*, 29.
- Yao, B. Z., Yang, X., Lin, L., Lee, M. W., & Zhu, S.-C. (2010). I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8).
- Yao, T., Pan, Y., Li, Y., & Mei, T. (2017). Incorporating copying mechanism in image captioning for learning novel objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6580–6588).
- Yao, T., Pan, Y., Li, Y., & Mei, T. (2018). Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 684–699).
- Yao, T., Pan, Y., Li, Y., & Mei, T. (2019). Hierarchy parsing for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 2621–2629).
- Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., Wang, L., Huang, Q., Wang, Y., Mahoney, M., et al. (2021). Hawq-v3: Dyadic neural network quantization. (pp. 11875–11886).
- Zhang, L., Sung, F., Feng, L., Xiang, T., Gong, S., Yang, Y., & Hospedales, T. (2017). Actor-critic sequence training for image captioning. In *Visually-Grounded Interaction and Language (ViGIL): NIPS 2017 Workshop*.
- Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., & Gao, J. (2021a). Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5579–5588).
- Zhang, X., Sun, X., Luo, Y., Ji, J., Zhou, Y., Wu, Y., Huang, F., & Ji, R. (2021b). Rstnet: Captioning with adaptive attention on visual and non-visual words. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15465–15474).
- Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems* (pp. 649–657).
- Zhenxian, L., Feirong, F., Xiaobao, Y., & Chen, D. (2021). An efficient image captioning method based on generative adversarial networks. In *2021 4th International Conference on Artificial Intelligence and Pattern Recognition* (pp. 374–379).
- Zhong, Y., Wang, L., Chen, J., Yu, D., & Li, Y. (2020). Comprehensive image captioning via scene graph decomposition. In *European Conference on Computer Vision* (pp. 211–229). Springer.
- Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., & Gao, J. (2020). Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34 (pp. 13041–13049).

List of Scientific Publications by the Author on the Topic of the Dissertation

Papers in the Reviewed Scientific Journals

Atliha, V., & Šešok, D. (2020b). Text augmentation using BERT for image captioning. *Applied Sciences*, 10(17): 5978. <https://doi.org/10.3390/app10175978>

Atliha, V., & Šešok, D. (2022a). Image-Captioning Model Compression. *Applied Sciences*, 12(3): 1638. <https://doi.org/10.3390/app12031638>

Papers in Other Editions

Atliha, V., & Šešok, D. (2020a). Comparison of VGG and ResNet used as Encoders for Image Captioning. In *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1–4). IEEE. <https://doi.org/10.1109/eStream50540.2020.9108880>

Atliha, V., & Šešok, D. (2021). Pretrained Word Embeddings for Image Captioning. In *2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1–4). IEEE. <https://doi.org/10.1109/eStream53087.2021.9431465>

Atliha, V., Šešok, D., & Valentinavičius, S (2022b). Sequence-level knowledge distillation for image captioning model compression. In *VII International scientific conference "High Technologies. Business. Society"*, Vol. 7 (pp. 7–9). Sofia: Scientific Technical Union Of Mechanical Engineering Industry-4.0. <https://stumejournals.com/journals/i4/2022/1/7>

Atliha, V., & Šešok, D. (2022c). Text Augmentation for Compressed Image Captioning Models. In *2022 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1–4). IEEE. <https://doi.org/10.1109/eStream56157.2022.9781641>

Summary in Lithuanian

Įvadas

Problemos formulavimas

Vaizdo aprašų generavimas yra automatinio tekstinio vaizdo aprašymo generavimo procesas. Šio uždavinio tikslas – sukurti aprašymus, kurie būtų kuo labiau panašesni į tai, kaip žmogus apibūdintų matomą vaizdą. Pastaraisiais metais dauguma vaizdų antraščių generavimo uždavinio sprendimo būdų siejami su neuroninių tinklų naudojimu, o dauguma naudojamų architektūrų yra kodavimo ir-dekodavimo tipo.

Naujausiuose šios srities tyrimuose dėmesys telkiamas į neuroninių tinklų architektūrų modifikavimą. Tačiau gerėjant modelių kokybei, jų dydis taip pat didėja, todėl tampa sudėtinga naudoti naujausius modelius vaizdų antraštėms sudaryti ribotus išteklius (ypač atminties) turinčiuose įrenginiuose, pvz., mobiliuosiuose ar nešiojamuosiuose, kur potencialiai yra daug pritaikymo galimybių. Todėl svarbu ir toliau gerinti vaizdų antraščių generavimo sistemų kokybę nedidinant jų dydžio. Taip pat aktualu mažinti sistemų dydį neprarandant kokybės, kad šios sistemos būtų praktiškesnės.

Ši disertacija visų pirma orientuota į pastarojo tikslo siekimą, pagerinti vaizdų antraščių generavimo sistemų kokybę nedidinant jų dydžio, taip pat sumažinti sistemų dydį, neprarandant geros kokybės.

Darbo aktualumas

Vaizdų antraščių generavimas yra svarbi užduotis kompiuterinio regėjimo srityje, turinti daug praktinių pritaikymų, pavyzdžiui, pagalba regėjimo negalia turintiems asmenims, pagalba ieškant vaizdų, geresnis vaizdų apdorojimas dirbtinio intelekto sistemose.

Atminties vartojimo sumažinimas neprarandant kokybės yra labai svarbus vaizdų antraščių generavimo modelio kūrimo aspektas, ypač ribotos atminties įrenginiuose, tokiuose kaip išmanieji telefonai ir įterptosios sistemos. Sprendžiant šį iššūkį galima diegti vaizdų antraščių generavimo modelius ribotos atminties įrenginiuose, kad jie taptų prieinamesni didesniai vartotojų ratui.

Be to, vaizdų antraščių generavimo modelių kokybės gerinimas be didelių architektūros pokyčių taip pat yra labai svarbi tyrimų sritis. Nors sudėtingos architektūros ir metodai gali pagerinti modelių tikslumą, jie dažnai susiję su padidėjusiomis skaičiavimo sąnaudomis ir sudėtingumu. Todėl radimas būdų, leidžiančių pagerinti vaizdų antraščių generavimo modelių kokybę, išlaikant paprastą ir efektyvią architektūrą, gali padidinti šių modelių praktiškumą ir naudingumą.

Tyrimų objektas

Šios disertacijos tyrimo objektas yra duomenų rinkinio papildymo (angl. augmentation) metodai, skirti vaizdų antraščių generavimo modelių kokybei pagerinti, ir vaizdų antraščių generavimo modelių glaudinimo (angl. compression) metodai.

Disertacijos tikslas

Šiuo tyrimu siekiama pagerinti vaizdų antraščių generavimo sistemų kokybės metrikas ir sumažinti vaizdų antraščių generavimo modelių dydį neprarandant kokybės.

Disertacijos tikslai

Norint pasiekti disertacijos tikslą, buvo iškelti šie uždaviniai:

1. Pasiūlyti ir įvertinti modelių glaudinimo metodus, siekiant sumažinti vaizdų antraščių generavimo modelių dydį daugiau nei 80% be reikšmingo (daugiau nei 5%) kokybės praradimo.
2. Sukurti teksto papildymo (angl. augmentation) metodus, skirtus išplėsti esamus duomenų rinkinius ir pagerinti esamų vaizdų antraščių generavimo sistemų kokybę.

Tyrimų metodika

Šioje disertacijoje panaudota kiekybinė metodika.

Siekiant geriau suprasti vaizdų antraščių sistemų tobulinimo galimybes, buvo atlikta dabartinių metodų literatūros apžvalga.

Vėliau buvo pasiūlyti keli esamų modelių patobulinimai į nauji modelių glaudinimo ir kokybės gerinimo metodai. Buvo atliktas siūlomų metodų įvertinimas, siekiant nustatyti

geriausiai veikiančius mašininio mokymosi ir giliojo mokymosi metodus. Galiausiai atlikta metrikų analizė, siekiant padaryti išvadas apie siūlomų metodų efektyvumą.

Disertacijos mokslinis naujumas

Šio tyrimo mokslinis naujumas apibūdinamas taip:

- Buvo pasiūlyta naudoti giliojo mokymosi modelių glaudinimo metodus, tokius kaip hiperparametrų keitimas, apkarpymas (angl. pruning), kvantavimas (angl. quantization) ir žinių distiliavimas vaizdų antraščių generavimo modeliams.
- Buvo pasiūlytas vaizdų antraščių duomenų rinkinių papildymas naudojant sinoniminius pakaitalus ir kontekstualizuotus žodžių įterpimus.
- Vaizdų antraščių generavimo modeliui buvo pasiūlyta naudoti iš anksto paruoštus “Word2vec” ir iš anksto paruoštus “GloVe” įterpimus.
- Teksto papildymo būdai buvo derinami su glaudinimo metodais, kad būtų pasiektas dar geresnis našumas.

Darbo rezultatų praktinė reikšmė

Šio tyrimo išvados turi tiek teorinę, tiek praktinę reikšmę. Vaizdų antraščių generavimo modelio glaudinimo srityje šio tyrimo rezultatai sudaro tvirtą pagrindą tolesniems tyrimams ir rodo, kad dabartiniai naujausi modeliai nenaudoja savo dydžio taip efektyviai, kaip galėtų ir kad patobulinimai gali būti atlikti nedidinant šių modelių dydžio. Praktiniu požiūriu geriausi modeliai, gauti šiame tyrime, yra gana maži ir juos būtų galima naudoti įrenginiuose, kurių atmintis yra ribota, kas anksčiau nebuvo įmanoma. Suglaudinus vaizdų antraščių generavimo modelius, šios technologijos pranašumai tampa prieinami platesnei auditorijai, įskaitant žmones, kurie gali neturėti prieigos prie aukščiausios klasės kompiuterinių išteklių.

Be to, modelių glaudinimo metodai gali padėti sumažinti giliojo mokymosi modelių, kuriems apmokyti ir vykdyti reikia daug energijos, poveikį aplinkai. Padarę šiuos modelius efektyvesnius, mokslininkai gali padėti sumažinti AI pramonės anglies pėdsaką.

Įgyvendinus idėją naudoti teksto papildymą vaizdų antraštėms generuoti, mokslininkams atsiveria galimybė ištirti kitus papildymo būdus ir jų įtaką modelio kokybei. Jau buvo paskelbti keli naujausi darbai, kuriuose remiamasi šiuo tyrimu, o tai rodo didelį susidomėjimą šia tema.

Galiausiai, įvertinus vaizdo dekodierio ir iš anksto paruoštų žodžių įterpimų poveikį modelio kokybei, galima matyti, kad patobulinimus galima atlikti modifikuojant tik mažas modelio dalis, nereikalaujant sudėtingų naujų architektūrų ar mokymo metodų. Apibendrinant, šis tyrimas turi praktinę vertę, nes padeda tobulinti vaizdų antraščių generavimą ir padaryti šiuos modelius labiau prieinamus.

Ginamieji teiginiai

Oficialiomis gintinomis hipotezėmis gali būti šie teiginiai, pagrįsti disertacijos rezultatais:

1. Giliojo mokymosi modelių glaudinimo technikų (pvz., apkarpymo, kvantavimo ir žinių distiliavimo) naudojimas vaizdų antraščių generavimo modeliams disertacijoje siūlomame derinyje leidžia sumažinti modelio dydį iki 91% prarandant kokybės metriką iki 3%.
2. Vaizdų antraščių duomenų rinkinių papildymai, naudojant sinoniminius pakaitalus ir kontekstualizuotus žodžių įterpimus, ypač skirtus naudoti vaizdų antraščių generavimo sistemose, pagerina kokybę iki 5%, priklausomai nuo modelio.
3. Disertacijoje pasiūlytas geresnių dekodierių ir iš anksto paruoštų žodžių įterpimų naudojimas gali pagerinti vaizdų antraštės metodų kokybę iki 5%.

Darbo rezultatų aprobavimas

Disertacijos rezultatai paskelbti šešiuose moksliniuose leidiniuose. Du straipsniai buvo paskelbti nurodytuose *Clarivate Analytics* (taip pat vadinamuose *Thomson Reuters*) *Web of Science* duomenų bazių žurnaluose. Keturi straipsniai buvo paskelbti konferencijų pranešimų straipsnių rinkiniuose. Autorius skaitė 4 pranešimus Lietuvoje ir užsienyje:

- 2020 *IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2020, Vilnius, Lietuva;
- 2021 *IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2021, Vilnius, Lietuva;
- VII *International scientific conference „High Technologies. Business. Society“*, 2022, Borovets, Bulgarija;
- 2022 *IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 2022, Vilnius, Lietuva.

Disertacijos struktūra

Mokslinį darbą sudaro disertacijos įvadas, trys pagrindiniai skyriai, bendrosios išvados, literatūros sąrašas, autoriaus publikacijų sąrašas.

Pirmasis skyrius yra apie vaizdų antraščių generavimo sistemų kūrimą. Jis apima skirtingus metodus, tipines architektūras, bendrus vaizdinės kalbos metodus, mokymo metodus, duomenų rinkinius ir vertinimo metrikas.

Antrajame skyriuje dėmesys sutelkiamas į vaizdų antraščių generavimo neuroninių tinklų modelių dydžio sumažinimą neprarandant kokybės.

Trečiasis šios disertacijos skyrius yra apie vaizdų antraščių generavimo modelių našumo pagerinimą nedarant esminių jų architektūros pakeitimų.

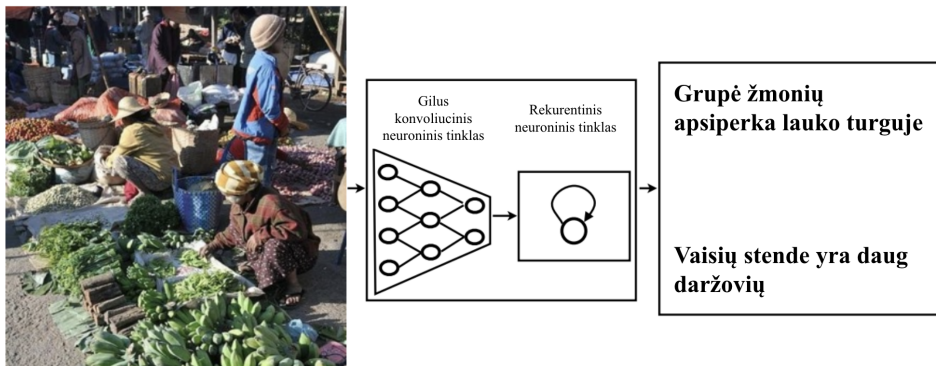
Bendra disertacijos apimtis – 123 puslapiai be priedų.

Tekste yra 56 lygtys, 34 paveiksleliai ir 28 lentelės. Disertacijos tekste buvo panaudotos 169 literatūros sąrašo nuorodų.

1. Vaizdų antraščių metodų literatūros apžvalga

Vaizdų antraščių generavimo užduotis yra kompiuterinio matymo (CV) ir natūralios kalbos apdorojimo (NLP) sankirtoje. Klasikinių vaizdų antraščių generavimo sistemų tikslas sukurti panašius į žmogaus kuriamus tekstinius aprašymus, pagrįstus vaizdine įvestimi. Kadangi vaizdų antraštės sujungiamos ir vaizdinėje, ir tekstinėje srityje, vaizdų antraščių generavimas yra sudegingesnis nei įprastos domeno užduotys CV (pvz., klasifikavimas ir aptikimas) arba NLP (pvz., nuotaikų klasifikavimas ir vertimas). Tai reiškia, kad aukštos kokybės vaizdų antraštės turėtų būti sudarytos iš aukštos kokybės vaizdo ir teksto posistemų, tačiau tai taip pat reiškia, kad tokioms sistemoms gali būti naudingi abiejų sričių tyrimai. Kitaip sakant, vaizdų antraščių generavimo užduotis gali būti vertinama kaip kalbos vertimo užduotis, kai įvesties „kalba“ yra vaizdinis turinys.

Kodavimo ir dekodavimo sistema yra populiariausias būdas kurti vaizdų antraštes nuo pradžios iki galo. Šią sistemą išpopuliarino vienas daugiausia cituojamų darbų vaizdų antraščių generavimo srityje Vinyals et al. (2015) (S1.1 pav.).



S1.1 pav. Kodavimo ir dekodavimo architektūra vaizdai aprašyti (Vinyals et al., 2015)

Tipiška kodavimo ir dekodavimo sistema susideda iš dviejų dalių: vizualinio supratimo modelio, kuris veikia kaip koduotuvas, ir kalbos modelio, kuris veikia kaip dekoderis. Vaizdinis modelis paprastai įgyvendinamas kaip didelis konvoliucinis neuroninis tinklas, kuris gali suglaudinti paveikslą į mažo matmens vaizdą. Kalbos modelis paprastai įgyvendinamas kaip pasikartojantis neuroninis tinklas, kuris gali generuoti paketų seką, pagrįstą paveikslo atvaizdavimu.

Vienas iš labiausiai paplitusių vaizdo antraštės modelių didelis konvoliucinis neuroninis tinklas (CNN), naudojamas kaip dekoderis, paverčiantis vaizdą į fiksuoto dydžio įterpimą žemesnio matmens erdvėje:

$$a = E(I), \quad (S1)$$

kur E yra CNN.

Kalbos modeliai yra esminė bet kokios vaizdų antraščių generavimo sistemos dalis. Pagrindinis jų tikslas – įvertinti tikimybę, kad natūralioje kalboje atsiras žodžių seka. Remiantis

šiuo modeliu, galima sukurti žodžių generatorių, kuris tinka vaizdų antraščių generavimo sistemos dekodavimo daliai. Todėl nenuostabu, kad vaizdų antraščių generavimo užduotyse dekoderiui naudojamos tipinės kalbos modelių architektūros.

Vienas iš svarbių vaizdų antraščių generavimo modelių mokymo aspektų yra tinkamų praradimo funkcijų ir mokymo strategijų pasirinkimas. Praradimo funkcijos naudojamos skirtumui tarp modelio prognozuojamos išvesties ir tikrų duomenų antraščių matuoti, taip pat modelio parametrams atnaujinti treniruočių metu.

Viena dažna praradimo funkcija, naudojama vaizdų antraštėms generuoti, yra kryžminės entropijos praradimas. Šis praradimas yra įprasta praradimo funkcija, naudojama atliekant prižiūravimo mokymosi užduotis, įskaitant vaizdų antraščių generavimą. Vaizdų antraščių generavimo tikslas išmokyti modelį, kad jis sukurtų vaizdo antraštės natūralia kalba. Kryžminės entropijos praradimo funkcija naudojama skirtumui tarp modelio numatytų antraščių ir tikrų duomenų antraščių matuoti ir naudojama modelio parametrams atnaujinti treniruočių metu.

Kryžminės entropijos praradimo funkcija apskaičiuojama kaip neigiama tikrųjų antraščių log tikimybė, atsižvelgiant į numatomas tikimybes. Šią praradimo funkciją galima parašyti taip:

$$L = - \sum_{i=1}^N y_i \log \hat{y}_i, \quad (S2)$$

kur y_i yra pagrindinė tikroji antraštė i -ajam pavyzdžiui, o $\hat{y}_i$ yra i -ojo pavyzdžio numatoma tikimybė.

Norint išmokyti ir įvertinti vaizdų antraščių generavimo modelius, būtina turėti prieigą prie didelio masto duomenų rinkinių, kuriuose yra įvairių vaizdų ir juos atitinkančių antraščių. Vienas iš dažniausiai naudojamų duomenų rinkinių vaizdų antraštėms generuoti yra "Microsoft" COCO duomenų rinkinys aprašytas straipsnyje Lin et al. (2014). Rinkinyje iš viso yra daugiau nei 200 000 vaizdų, kurių kiekvienas komentuojamas keliomis antraštėmis, apibūdinančiomis vaizduose esančius objektus ir scenas. Duomenų rinkinys sukurtas taip, kad jame būtų daug įvairių objektų ir scenų, įskaitant sudėtingus ir neįprastus objektus. Taip pat jame yra įvairus vaizdų fonų ir kontekstų rinkinys. Duomenų rinkinio antraštės parašytos natūralia kalba, todėl yra patikimas duomenų šaltinis mokant ir vertinant vaizdų antraščių generavimo modelius.

Yra keletas metrikų, kurios dažniausiai naudojamos vaizdų antraščių generavimo modelių veikimui įvertinti: BLEU, ROUGE, METEOR, CIDEr, SPICE. Šios metrikos skirtos sugeneruotų antraščių kokybei matuoti ir gali būti naudojamos skirtingų algoritmų ir modelių našumui palyginti.

CIDEr (angl. consensus-based image description evaluation) ir SPICE (angl. semantic propositional image caption evaluation) yra dvi dažniausiai naudojamos metrikos, skirtos vaizdų antraščių generavimo modelių veikimui įvertinti. CIDEr ir SPICE yra populiarios vaizdų antraščių generavimo metrikos, nes buvo įrodyta, kad jos efektyviau įvertina sugeneruotų antraščių kokybę, nei kitos metrikos, tokios kaip BLEU ir ROUGE. Pavyzdžiui, tyrime Vedantam et al. (2015) buvo nustatyta, kad CIDEr yra žymiai geresnė metrika nei BLEU ir ROUGE, vertinant vaizdų antraščių, sugeneruotų naudojant naujausius algoritmus, kokybę.

2. Atminties vartojimo mažinimas vaizdų antraščių neuroninių tinklų modeliuose

Koduotuvas yra esminis daugelio vaizdų antraščių generavimo modelių komponentas. Dažnai tai yra didžiausia ir sunkiausia modelio dalis, kurią realizuoja konvoliucinis neuroninis tinklas. Taigi kodavimo įrenginio glaudinimas yra vienas iš efektyviausių būdų sumažinti bendrą vaizdo antraščių generavimo modelio atminties suvartojimą. Mūsų tyrimas parodė, kad pakeitus kodavimo įrenginį iš standartinio “Faster R-CNN” su “ResNet101” magistrale į “EfficientDet-D0”, modelio dydis efektyviai sumažinamas, o kokybės rodikliai išlieka maždaug to paties lygmens.

Antrasis taikytas metodas yra dekoderio hiperparametrų derinimas. Pristatytas mastelio koeficientas γ ir ištirta jo įtaką modelio veikimui. Skirtingų modelių jis buvo skirtingas, tačiau γ reikšmės apie 0.25–0.5 užtikrina gerą atminties ir kokybės pusiausvyrą.

Kitas metodas, vėliau pritaikytas dekodavimui, yra apkarpymas. Pagal mūsų eksperimentus l_1 apkarpymas iš išbandytų metodų veikia geriausiai. Optimalus apkarpymo koeficientas taip pat priklauso nuo konkretaus modelio ir gali svyruoti nuo 0.1 iki 0.5 bei padeda sumažinti modelio dydį.

Paskutinis metodas, kuris buvo pritaikytas dekodavimui glaudinti, yra kvantavimas, padedantis padeda sumažinti modelio dydį beveik nepakeičiant kokybės metrikos.

Bendra kombinuoto metodo schema parodyta 6 paveiksle.

Alternatyvus mūsų pasiūlytas metodas – užuot keitus hiperparametrų apkarpius bei kvantavus, naudoti žinių distiliavimą dekoderiui glaudinti. Metodas suteikė dar efektyvesnį modelio glaudinimo būdą, tačiau šiek tiek sumažino kokybę. Metodas pateiktas 7 paveiksle.

S2.1 lentelėje pateikiamas galutinis visų skyriuose siūlomų glaudijimo metodų ir kitų autorių metodų palyginimas. “Originalus modelis” reiškia originalų modelį be jokio suspaudimo metodo. “Pirmasis metodas” yra geriausias kodavimo ir dekoderio architektūrinių pakeitimų, apkarpymo ir kvantavimo metodų derinys: EfficientDet-D0 kodavimo įrenginys, dekoderis su mastelio koeficientu 0,5, apkarpymo koeficientai 0,1 ir 0,5 Up-Down ir AoANet modeliams ir atitinkamai pritaikytas kvantavimas. “Antrieji metodai” yra geriausias kodavimo įrenginio architektūros pakeitimo, mažesnio dekoderio ir geriausio žinių distiliavimo metodo derinys: EfficientDet-D0 kodavimo įrenginys, dekoderis su skalės koeficientu 0,25, žinių distiliavimas su 10 ir 3 pluošto dydžiais, skirta Up-Down ir AoANet modeliams atitinkamai. Matyti, kad “Pirmasis metodas” yra geriausias pagal glaudinimo efekto ir metrikos kitimo santykį.

3. Neuroninių tinklų modelių vaizdų antraščių kokybės gerinimas

Vaizdų antraščių generavimo modeliai, nepaisant jų aukštos kokybės, turi ribotą galimybę generuoti įvairius objektų aprašymus įvairiose situacijose. Šis apribojimas daugiausia susijęs su tuo, kad dabartiniai mokymo duomenų rinkiniai, nors ir yra dideli pagal vaizdų skaičių, turi palyginti mažą antraščių skaičių vienam vaizdui. Siekiant išspręsti šią problemą, mokymo duomenų rinkinys išplėstas taikant teksto papildymo metodus, įskaitant sinonimų vartojimą kaip pagrindą ir moderniausio kalbos modelio BERT taikymą skaičiuojant conceptualizuotus žodžių įterpimus. Šie metodai leido padidinti mokymo duomenų įvairovę,

S2.1 lentelė. Visų glaudinimo metodų pagrindinių metrikų vertinimo rezultatai. Stulpelis “Size” (Dydis) nurodo visą modelį

	B@4	METEOR	ROUGE-L	CIDEr	SPICE	Size
Up-Down						
Originalus modelis	36.3	27.7	56.9	120.1	21.4	518.5 MB
Pirmasis metodas	35.3	26.9	56.3	118.4	20.8	58.5 MB
Antrieji metodai	35.4	26.9	56.3	116.6	20.5	22.3 MB
AoANet						
Originalus modelis	38.9	29.2	58.8	129.8	22.4	553.2 MB
Pirmasis metodas	38.3	29.0	58.3	127.4	21.4	34.8 MB
Antrieji metodai	37.5	28	57.5	124.5	21.5	24.3 MB

pagerindami vaizdų antraščių generavimo modelio galimybes generuoti platesnį aprašymų spektrą.

Skirtingai nuo vaizdo ir kalbos apdorojimo, kai atsitiktinio triukšmo pridėjimas prie įvesties signalo gali būti naudinga papildymo technika, šis metodas netinka tekstui papildyti. Santykinė raidžių tvarka ir jų buvimas žodyje gali labai paveikti paties žodžio semantinę reikšmę, todėl paprasčiausiai atsitiktinio triukšmo įtraukimas į įvesties tekstą greičiausiai nebus efektyvus. Geriausias būdas papildyti tekstą perfrazuoti sakinius taip, kaip tai darytų žmogus, tačiau šį metodą sunku įgyvendinti ir pritaikyti dideliems mokymo duomenų rinkiniams. Vienas paprastesnis, bet vis tiek veiksmingas metodas yra sinoniminis pakeitimas, pirmą kartą pristatytas natūralios kalbos apdorojimo užduočiai Zhang et al. (2015).

Kontekstinis žodžių įterpimo metodas, aprašytas dokumentuose Fadaee et al. (2017) ir Kobayashi (2018), gali būti naudojamas tam tikram sakinių, apibūdinančių vaizdą, rinkiniui papildyti. Tiksliau, atsižvelgiant į vaizdą I ir sakinių $C = c_1, \dots, c_k$ rinkinį, apibūdinantį tą vaizdą, kiekvienas sakiny yra žodžių $c_i = (w_{i,1}, w_{i,2}, \dots, w_{i,l_i})$.

Bendrą papildymo metodo schemą galima rasti 8 pav.

Tyrimuose palyginti penki skirtingi papildymo būdai, kaip papildyti vaizdų antraščių duomenų rinkinį. Papildymas naudotas tik mokymo duomenų rinkinio daliai, o palyginimui naudotas modelis be jokio papildymo. Kituose metoduose duomenų rinknys papildytas naudojant BERT su padidinimo koeficientais d , lygiais 2 ir 3, taip pat sinonimais su padidinimo koeficientu 2. Visais atvejais kaip numatyta, pakeitimo norma p buvo nustatyta 0,1, išskyrus pakeitimo normos poveikio tyrimą.

Norint įvertinti šių papildymo metodų efektyvumą, papildytuose duomenų rinkiniuose parengtas naujausias modelis (Cornia et al., 2020b). Atlikti išsamūs eksperimentai, siekant nustatyti geriausią šio modelio padidinimo metodą. Rezultatus palyginti su kitais moderniausiais modeliais, pvz., Huang et al. (2019a) ir Pan et al. (2020), naudojant geriausią duomenų rinkinio variantą, pasirinktą remiantis ankstesniais eksperimentais su (Cornia et al., 2020b). Visi modeliai buvo mokomi naudojant atvirųjų šaltinių kodą, kurį išleido atitinkamų straipsnių autoriai. Papildymui naudojome nlpaug biblioteką. Svarbu tai, kad naudojant atvirųjų šaltinių kodą visi naudojami modeliai pasirodė šiek tiek prastesni nei aprašyti atitinkamuose straipsniuose.

Galutiniai visų apmokytų (Cornia et al., 2020b) modelių testų rezultatai apibendrinti S3.1 lentelėje. Modelis, parengtas naudojant du kartus padidintą duomenų rinkinį, gautą naudojant BERT su $p = 0,1$ padidinimu, rodo geriausius rezultatus beveik visose metrikose

ir gerokai pranoksta modelį, parengtą pagal pradinį duomenų rinkinį: 2,7 taško CIDEr ir 0,2 taško SPICE. Tai rodo siūlomo papildymo metodo veiksmingumą gerinant vaizdų antraščių generavimo modelių kokybę. Papildymas gali būti naudojamas kaip paprastas ir efektyvus būdas pagerinti esamų moderniausių metodų veikimą nekeičiant pagrindinio modelio.

S3.1 lentelė. Cornia et al. (2020b) modelio, parengto taikant siūlomus didinimo metodus, vertinimo rezultatai

Padidinimo tipas	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
Jokio padidinimo	38,3	28,6	58,0	126,1	22,6
BERT, $d = 2$, $p = 0.1$	37,8	28,9	58,3	128,8	22,8
BERT, $d = 2$, $p = 0.3$	37,3	28,5	57,8	125,4	21,9
BERT, $d = 2$, $p = 0.5$	37,3	28,8	58,0	126,5	22,3
BERT, $d = 3$	37,9	28,6	57,9	127,2	22,4
Sinonimai, $d = 2$	37,7	28,7	57,8	127,4	22,2

S3.2 lentelėje apibendrinami visų trijų modelių, apmokytų naudojant BERT papildymą su $d = 2$ ir $p = 0.1$, rezultatai ir palyginami su skirtingų autorių rezultatais. Rezultatai rodo, kad visi modeliai, apmokyti naudojant papildymą, lenkia atitinkamus modelius, taikytus be papildymo. Tai patvirtina išvadas apie siūlomo papildymo metodo naudą mokant moderniausių vaizdų antraščių generavimo modelius. Tai rodo, kad mokymo duomenų papildymas naudojant BERT gali būti veiksmingas būdas pagerinti modelio veikimą.

S3.2 lentelė. Visų trijų išbandytų modelių, apmokytų mūsų padidinimo metodais, vertinimo rezultatai

Modelis	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
M^2 Transformers, no padidinimo	38,3	28,6	58,0	126,1	22,6
M^2 Transformers, padidinimo	37,8	28,9	58,3	128,8	22,8
AoANet, jokio padidinimo	38,2	28,6	57,9	125,9	22,3
AoANet, padidinimo	37,6	28,8	58,0	128,4	22,5
X-LAN, jokio padidinimo	38,8	28,9	58,2	126,1	22,8
X-LAN, padidinimo	37,9	29,2	58,3	128,6	22,9

Bendrosios išvados

1. Literatūros apžvalga parodė didelį tyrėjų susidomėjimą vaizdų antraščių generavimu. Įprastas vaizdų antraščių generavimo modelis turi kodavimo-dekoderio architektūrą, kur koduotuvas paprastai yra vaizdo modelis (pvz., VGG arba ResNet), o ir dekodavimo paprastai yra kalbos modelis (pvz., RNN arba LSTM). Siekiant pagerinti šių modelių kokybę, dažnai naudojamos skirtingos architektūros neatsižvelgiant į jų poveikį atminties suvartojimui. Daugumoje straipsnių naudojami tie

patys duomenų rinkiniai (pvz., MS COCO ir “Flickr30k”) su tais pačiais išankstinio apdorojimo metodais. Tačiau šių duomenų rinkinių išplėtimo klausimas lieka atviras.

2. Siūlomi nauji vaizdų antraščių generavimo modelių glaudinimo metodai leidžia sumažinti modelio dydį iki 91,2% prarandant tik 1,4% CIDEr balą ir 2,8% SPICE balą. Geriausias bendras metodas yra naudoti “EfficientDet-D0” kodavimo įrenginį, dekoderį, kurio mastelio koeficientas yra 0,5, apkarpytą su koeficientu, priklausančiu nuo modelio architektūros, ir naudoti kvantavimą. Šis metodas leidžia pasiekti kokybės rodiklius, palyginamus su kitais autoriaus metodais, naudojant žymiai mažiau atminties.
3. Buvo pasiūlyta naudoti teksto papildymo metodus, įskaitant sinonimų vartojį ir kontekstualizuotą žodžių įterpimą. Mažesnis pradinės antraštės pakeitimo koeficientas (0,1) su BERT sugeneruotais sakiniais pagerina našumą, o didesnis pakeitimo koeficientas (0,5) našumą mažina. Duomenų rinkinio papildymas naudojant BERT taip pat pagerina modelio našumą, palyginti su sinonimų papildymu arba pradinio duomenų rinkinio naudojimu. Šio papildymo metodo veiksmingumas yra ribotas, o mokymas naudojant duomenų rinkinį, kuris buvo padidintas tris kartus, lemia prastesnį našumą, palyginti su du kartus padidintu duomenų rinkiniu. Modelis, parengtas naudojant du kartus padidintą duomenų rinkinį naudojant BERT su 0,1 pakeitimo koeficientu, 3,4% pralenkia modelį, parengtą pagal pradinį duomenų rinkinį CIDEr ir 5% SPICE.

Viktar ATLIHA

IMPROVING IMAGE CAPTIONING METHODS
USING MACHINE LEARNING APPROACHES

Doctoral Dissertation

Technological Sciences,
Informatics Engineering (T 007)

VAIZDŲ ANTRAŠČIŲ GENERAVIMO
METODŲ TOBULINIMAS MAŠININIO
MOKYMOSI METODAIS

Daktaro disertacija

Technologijos mokslai,
Informatikos inžinerija (T 007)

Anglų kalbos redaktorė Jūratė Griškėnaitė
Lietuvių kalbos redaktorė Rita Malikėnienė

2023 05 11. 10 sp. l. Tiražas 20 egz.
Leidinio el. versija <https://doi.org/10.20334/2023-021-M>
Vilniaus Gedimino technikos universitetas
Saulėtekio al. 11, 10223 Vilnius
Spausdino UAB „Ciklonas“,
Žirmūnų g. 68, 09124 Vilnius