The Optimal Choice of the Encoder-Decoder Model Components for Image Captioning

Mateusz Bartosiewicz and Marcin Iwanowski *

Posted Date: 14 August 2024

doi: 10.20944/preprints202408.1045.v1

Keywords: image captioning; image processing; image analysis; computer vision; recurrent neural network

Article

The Optimal Choice of the Encoder-Decoder Model Components for Image Captioning [†]

Mateusz Bartosiewicz and Marcin Iwanowski

Institute of Control and Industrial Electronics, Warsaw University of Technology, ul. Koszykowa 75; 00-662 Warszawa POLAND

[*] Correspondence: mateusz.bartosiewicz.dokt@pw.edu.pl

[†] This paper is an extended and modified version of our paper published in the FedCSIS 2023 conference Warsaw, Poland, 17–20 September, 2023

Abstract: Image captioning aims at generating meaningful verbal descriptions of a digital image. This domain is rapidly growing due to the enormous increase in available computational resources. The most advanced methods are, however, resource-demanding. In our paper, we return to the encoder-decoder deep learning model and investigate how replacing their component with newer equivalents improves overall effectiveness. The primary motivation of our study is to get the highest possible level of improvement of classic methods, which are applicable in less computational environments where most advanced models are too heavy to be efficiently applied. We investigate image feature extractors, recurrent neural networks, word embedding models, and word generation layers and discuss how each component influences the captioning model's overall performance. Our experiments are performed on the MS COCO 2014 dataset. As a result of our research, replacing components improves the quality of generating image captions. The results help design efficient models with optimal combinations of their components.

Keywords: image captioning; image processing; image analysis; computer vision; recurrent neural network

---

1. Introduction

Image captioning is an essential branch of modern artificial intelligence (AI). It aims to produce a verbal description of a digital image, usually as a single phrase. It combines computer vision and natural language processing, two significant fields in AI. Image captioning presents challenges such as understanding context, recognizing details, and generating coherent and relevant descriptions. It helps make visual content accessible to visually impaired people. By providing descriptive captions for images, AI can help them understand and interact with digital content more effectively. It can also automatically scan and describe large volumes of visual content, assisting in identifying inappropriate or harmful content on social media and other platforms. It improves user experience in various applications like photo organizing and searching. Automatically generated captions can help users find specific images without manually tagging every photo. Image captioning can enhance online catalogs, improve customer experience on retail sites, and provide better insights from visual data like social media trends and consumer behavior. In digitizing cultural and historical archives, AI-powered image captioning can help catalog and describe large volumes of visual data, making them more searchable and understandable.

As a result, image captioning is experiencing intense development nowadays. It is driven by developing deep learning methods made possible by increasing computing power. More and more new models are continuously appearing. Novel methods are resource-consuming; they require powerful computing servers to learn successfully and perform efficient inference processes. A drastic increase in complexity often obtains a slight increase in efficiency. Most recent transformer models are computationally intensive, requiring significant hardware and energy resources for training and inference. The high computational cost is due to the model size and the complex attention mechanisms that scale quadratically with the input length. This complexity necessitates extensive hardware, such as TPUs or multi-GPU setups, to efficiently manage the memory and processing requirements. Transformer models also demand large and diverse datasets for effective training. This extensive data requirement ensures that the model can learn a wide array of patterns and nuances in the data, which happens in the case of image captioning. However, acquiring and annotating such large datasets is costly and needs additional computational efforts. Furthermore, the quality of the training data directly impacts the model's performance, as any biases or errors in the data can be amplified in the model's outputs. On the other hand, older models, much less complex, are readily available. They consist of multiple components that their more advanced and efficient substitutes may replace. Finally, such a substitution could improve the overall efficiency of models.

Our paper thoroughly investigates the encoder-decoder architecture for image captioning, which consists of an encoder that converts the image into a feature vector and a decoder that decodes it as a single phrase describing the image's content. In our experiments, we make use of the classic approach, where the encoder employs the backbone model based on the convolutional neural network (CNN) that extracts features from the image. The decoder, in turn, is the recurrent neural network (RNN) that produces, word-by-word, an output sentence describing the image content. In addition, the captioning model includes supplementary components. The first is word embedding, which converts the one-hot-encoded word (vocabulary index) into a fixed-length vector that makes the RNN input. The second is the *Adaptation component*, which reduces and generalizes the image feature vector. Finally, the third one is responsible for merging the output of the RNN with the image feature vector and processing it by fully connected layers to get the prediction of the next word.

We focus our experiments on choosing particular components of the encoder-decoder captioning model and their influence on its efficiency. We aim to enhance the well-known encoder-decoder image captioning architecture by manipulating parameters in the model. By experimenting with CNN extractors, we captured more nuanced visual features, improving the model's understanding of complex scenes. Furthermore, integrating more sophisticated language models (Glove, FastText) can significantly enhance the generated captions' linguistic quality and semantic coherence. Our approach involves systematically tuning hyperparameters and expanding the various image and text feature extractors. These improvements collectively contribute to higher performance without main changes in the leading architecture, demonstrating the potential of our approach to bust the already existing image captioning model.

The paper is organized as follows. Section 2 contains a short survey of the literature. Section 3 describes the model under study and its components. In section 4, the training and evaluation processes are described. The results are reported in section 5. Section 6 concludes the paper.

2. Deep Learning Captioning Models

The history of image captioning, a field that merges computer vision and natural language processing, has evolved significantly over the years. The methods combine text and visual data and belong to the multimodal machine-learning approaches [1–3]. Image captioning has been a field of active research and development, showcasing the rapid advancements in computer vision and natural language processing. It continues to evolve, with ongoing research focusing on improving accuracy, context awareness, and the ability to generate more detailed and nuanced captions. The history of captioning can be divided into several periods. Initial attempts at image captioning were rudimentary and primarily rule-based. These systems relied on manually crafted rules for image recognition and text generation. Due to some uncertainty level, among other techniques, the fuzzy logic approach has been used [4,5]. In [6–8], authors applied fixed templates with blank slots filled with various objects, descriptive tokens, and situations extracted from images by the object detection systems. On the other hand, in [9], authors used already existing, predefined sentences. They created space of meaning from image features and compared images with sentences to find the most appropriate ones. Despite semantic and grammatical correctness, captions from traditional methods often differ from the way a human describes the image content. This period was characterized by limited success due to the time constraints of the technology available.

In early 2000, researchers began to train deep learning models to recognize image patterns and generate descriptions [10]. However, these models still needed to improve their capabilities, often requiring hand-crafted features and extensive domain knowledge. The breakthrough came in the 2010s with the application of deep learning, where convolutional neural networks (CNN) [11] and recurrent neural networks (RNN) were combined. CNN focused on extracting features from images, while RNNs effectively handled sequential data, such as phrases consisting of consecutive words. Developing end-to-end trainable models in the mid-2010s, where a single model could be simultaneously trained on image processing and caption generation tasks, was a significant milestone [12]. It was enabled by large datasets like MS COCO (Microsoft Common Objects in Context) 2014, which provided extensive image and caption pairs for training.

Scenes in MS COCO 2014 dataset are complex and consist of animals and people in their everyday activities. Sentences are correct semantically and grammatically represent different aspects of an image. On the contrary, BreakingNews [13] and GoodNews [14] contain novel captions, therefore being taken from news articles. Conceptual Captions [15,16] are collections of 3.3 million and 12 million automatically collected from the internet, with weakly associated descriptions. Accordingly to the quantity, they are useful for the pre-training, but the source is not guaranteed according to the URL source. Therefore, during our experiments, we utilized MS COCO 2014 dataset, high-quality captions and contextual richness make it well-suited for training image captioning models. It also works as a standard benchmark of image captioning research, which helps in the consistent evaluation and advancement

Evaluation of image captioning methods is still a task that requires simulating human judgment. Due to that, it is still a complicated and comprehensive task. In pioneering work [17], authors suggested that neural networks can interpret deep semantics of images and word embedding. They proved that combined image features extracted by CNN and word embedding could hold semantic meaning. Inspired by the success of machine translation, [12] proposed using an encoder-decoder framework in image captioning, which has recently become dominant in the image captioning field. The paper [18] by Karpathy et al. introduced an architecture similar to human perception. The method generates novel descriptions over image regions, with R-CNN (Regional Convolutional Neural Networks) [19] for extracting image features and the RNN to generate consecutive words of caption iteratively. The model uses the multimodal embedding space to find the parts of the sentence that best fit the image regions. Differently from other proposed methods ([12,17,20]), where a global image vector was used, Karpathy focused on image regions, and a separate caption described each region. Finally, a spatial map generates the target word for image regions. These image captioning approaches, which focus on generating captions for each region of an image, are called dense image captioning [21–23].

The encoder-decoder architecture [12,24–26] considers the task of image captioning as the sequence-to-sequence problem. Using the image features extractor, the encoder encodes the image to the fixed-length vector. The most widely used are CNN networks such as VGG [27–29], ResNet [28,30,31] or Inception [32,33]. The decoder represents a language model in image captioning and generates natural language descriptions for the output. Most popular approaches used RNN [12,34–36].

There are two main approaches to controlling how the encoder and decoder are linked together [37]. The first, called injection, applies the image feature vector to initialize the state vector of the RNN [20,38–40]. The second approach is the merge architecture, where the image feature vector is combined with the output of the RNN. In our experiments, we use the latter approach, which proved to be more efficient [41–44].

The introduction of attention mechanisms has significantly improved the performance of the encoder-decoder image captioning models by enabling them to focus on different parts of an image when generating each word in the caption. Recently, transformer-based models have revolutionized image captioning. Vision Transformers (ViTs) [45] leverage self-attention mechanisms to process image patches, achieving competitive performance compared to CNN-based models. Method [46] employs an entire transformer architecture for both the encoder and decoder, eliminating convolutional layers.

In contrast, traditional CNN+transformer models typically use a CNN (e.g., ResNet) as the encoder to extract spatial features from images before passing them to a transformer decoder. The model's encoder is initialized with weights from VIT [45] model. In the [47], authors used X-linear attention block, with bilinear pooling to capture 2nd order interactions between elements during attention weights computation. It is achieved by simultaneously exploiting both spatial and channel-wise attention distributions. Furthermore, the X-linear attention block is integrated into the image encoder and sentence decoder to facilitate multi-modal reasoning.

This shift towards transformers has been further bolstered by large-scale vision-language pre-training paradigms, exemplified by models such as CLIP [48] and ALIGN [49], which use extensive datasets to pre-train on paired image-text data. Model [50] employs a shared multi-layer transformer network for fine-tuning encoding and decoding for image captioning tasks. The model is pre-trained using two unsupervised vision-language prediction tasks: bidirectional masked language prediction and sequence-to-sequence (seq2seq) masked language prediction. This dual approach enables the model to learn a more universal contextualized vision-language representation. By unifying the pre-training procedure, the model eliminates the need for separate models for encoding and decoding.

Moreover, transformer-based captioning models like OSCAR [51] and VinVL [52] encode visual features and textual context, producing contextually enriched captions. Model [52] improved the Transformer-based VL fusion model OSCAR by feeding it with a new object detection model and proved that visual features matter mostly in VL models. The latest model incorporates ResNeXt-152 C4 architecture, which is tailored for generating object-centric representations and detecting objects and their attributes, thanks to the attribute branch during fine-tuning.

Recent advancements in 3D representation learning have also led to significant innovations across 3D image captioning [53]. Model [54] enhances 3D point-cloud classification and segmentation, [55] address the aggregation and transfer of local features in point clouds, achieving robust performance in shape classification and segmentation, [56] leverages 3D point clouds for person re-identification, utilizing global semantic guidance and local feature extraction to achieve high accuracy while reducing model parameters. [57] advances scalable multimodal 3D representation learning by automatically generating language descriptions for 3D shapes, demonstrating state-of-the-art results in zero-shot classification, fine-tuning, and image captioning. Finally, [58] adapts GPT concepts to point clouds through an auto-regressive generation task. Collectively, these approaches highlight the potential of advanced neural network architectures and pre-training frameworks to enhance 3D data processing and image captioning. During the rapid development of image captioning methods, researchers also investigated other aspects of captions that are not just comparable to human judgment. Researchers focused on captions with a specific style. In [25], authors improved the descriptiveness of the generated captions by combining CNN and LSTM. In [59], authors focused on captions for visually impaired people. The developed model tends to create captions that describe the surrounding environment.

3. Preliminaries

3.1. Encoder-Decoder Image Captioning Model

In our paper, we investigate the merge approach to encoder-decoder-based captioning. The complete structure of the model under study is shown in Figure 1. The encoder-decoder merge model consists of two main components. The first component is responsible for encoding the visual information from the image. The CNN-based encoder processes the input image to extract visual features. The output of the encoder is a feature vector that represents the content of the image. Following most of the approaches [18,24,25,60–64], in our experiments, we use the pre-trained backbones. This convenient solution uses transfer learning to save time for learning image features (for details, see section 3.2). The continuous development of backbones results in the increasing efficiency of image feature extractors – the first part of our research focuses on their influence on the final captioning quality. In the data processing pipeline, the pre-trained feature extractor is followed by the *Adaptation* *component* that produces a shorter feature vector and – contrary to the pre-trained extractor – is trained during the training process, thus adapting the feature vector, generated by the off-the-shelf backbone, to the particular form of data used by the captioning model.

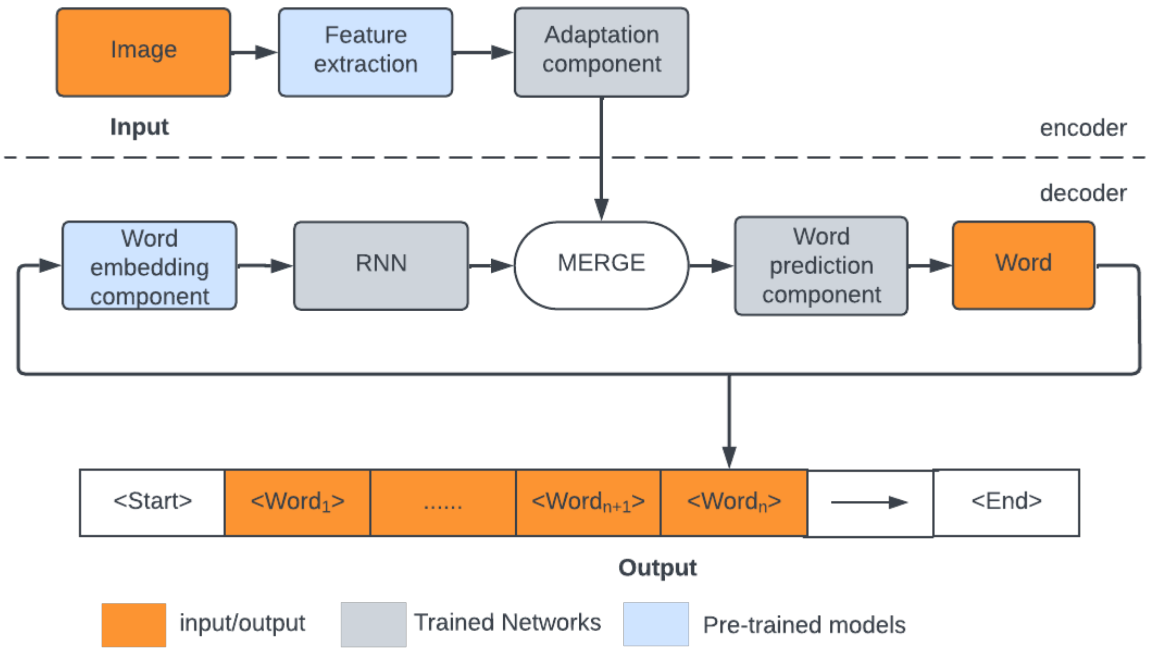


Figure 1. Encoder-decoder merge captioning model.

The decoder that generates a textual description is the second principal component of the captioning model. This component is based on a recurrent neural network. The RNN processes the feature vector sequentially, generating one word at a time and creating a coherent sentence. The decoder generates a caption, beginning with a *start* token. Word by word, the decoder predicts the following word in the sequence until it reaches an *end* token. The decoder learns to translate the visual features into meaningful sequences of consecutive words. In our experiments, we mainly use the LSTM model, but we also validate its alternative – the GRU recurrent network (for details, see section 3.3). As in many text generation models, we use, in addition, the word embedding models that present several advantages like dimensionality reduction and generalization properties. This issue is discussed in Section 3.4.

Linking the encoder and the decoder in the merge architecture is done by combining the image feature vector modified by the *Adaptation component* with the output of the RNN. There are two ways of doing this: vectors (adapted image features and output of the RNN) can be either added or concatenated. The result of this operation is next processed by the *Word prediction component*, composed of fully connected layers. It outputs a vector of size equal to the vocabulary size predicting the next word in a sentence in a one-hot encoded manner. On one hand, this word is added to the final sentence. On the other hand, it is used, along with previously generated words, as the decoder's recurrent input to predict the next word of the sentence. In our experiments, we validate various configurations of the captioning model's *Adaptation, Merging, and Word prediction components* (for details, see section 3.5).

During experiments, we checked how the model's choice of the above components affects the final results. We tested several configurations of *Adaptation, Merging, and Word prediction components* of the image captioning model.

3.2. Image Feature Extractors

Image features are essential in image captioning. Following the majority of approaches, we used pre-trained models- backbones in our experiments to extract them. Pretraining makes it possible to focus on the captioning model and restrict training to its remainder, leaving apart the tedious training of the image feature extractor. Weights of pre trained backbones are not updated during training but are used as the frozen layers of the image captioning model to extract specifics of the image. The vector of extracted features is the input to the *Adaptation component*, which updates weights during training. In our experiments [65], we selected several pre-trained feature extractors (backbones) among many available, focusing on ResNet, Inception, InceptionV3, Xception, DenseNet, and MobileNet. We also applied the older VGG backbone used in pioneering deep-learning captioning models for comparative purposes.

The VGG [66] is a group of convolutional neural networks widely used for image classification tasks. The most popular variants are VGG16 and VGG19. VGG16 consists of 13 convolutional and three dense layers and was trained to recognize 1000 object classes referring to objects depicted on input 224x224x3 color images from the ImageNet database [67]. The backbone network that produces the image feature vector of length 4096 has been obtained by cutting out the dense layers. VGG19 has three more CNN layers than VGG16. That allows us to learn richer data representations and achieve higher prediction results. On the other hand, VGG19 is more exposed to the vanishing gradient problem than VGG16 and requires more computational power.

The Resnet [68] network was created to support many layers while preventing the phenomenon of vanishing gradient in deep neural networks. The most popular variants are Resnet18, Resnet50, and Resnet100, where the number in the name represents the number of layers. The fundamental concept behind ResNet is the utilization of residual blocks, which introduce skip connections (shortcuts across layers). These connections allow the network to bypass specific layers, enabling it to learn residual mappings instead of directly attempting to approximate the desired underlying mapping.

Network structure consists of two main phases. In the beginning, the stack of skip connections layers is built. Those layers are omitted, and the activation function from the previous layer is used. In the next stage, the network is learned again, layers are expanded, and other parts of the network (residual blocks) learn deeper features of the image. Residual blocks are the heart of residual convolutional networks. They add skip connections to the network, which preserve essential elements of the picture till the end of the training, simultaneously allowing smooth gradient flow.

The Inception [69] model was created to deal with overfitting in very deep neural networks by going wider in layers rather than deeper. It is built among inception blocks that process input and repetitively pass the result to another inception block. Each block consists of four parallel layers 1x1, 3x3, 5x5, and max-pooling. 1x1 is used to reduce dimensions by channel-wise pooling. Thanks to that, the network can increase in depth without overfitting. Convolution is computed between each pixel and filter in the channel dimension to change the number of channels rather than the image size. 3x3 and 5x5 filters learn spatial features of the image in different scales and act similarly to human perception. Final max-pooling reduces the dimensions of the feature map. The most popular versions of the Inception network are Inception, InceptionV2, and InceptionV3.

The InceptionV3 [70] incorporated the best techniques to optimize and reduce the computational power needed for image feature extraction in the network. It is a deeper network than InceptionV2 and Inception, but its effectiveness was maintained. Also, auxiliary classifiers can be used to improve the convergence of very deep neural networks and combat the vanishing gradient problem. Factorized convolutions were used to reduce the number of parameters needed in the network, and smaller asymmetric convolutions allowed faster computations.

The Xception [71] is a variation of an Inception [69] model that decouples cross-channel and spatial correlations. Architecture is based on depthwise separable convolution layers and shortcuts between convolution blocks, as in Resnet. It consists of 36 convolutional layers divided into 14 modules. Each module is surrounded by residual connections, except the first and last modules. It has a simple and modular architecture and achieved better results than VGG16, Resnet and InceptionV3 in classical classification challenges.

DenseNet [72] was created to overcome the vanishing gradient problem in very long deep neural networks by simplifying data flow between layers. Architecture is similar to Resnet, but thanks to the simple change in connection between layers, DenseNet allows the reuse of parameters within the network and produces models with high accuracy. The structure of DenseNet is based on a stack of connectivity, transition, and bottleneck layers grouped in dense blocks. Every layer is connected with every other layer in a dense way. Dense block is a main part of DenseNet and reduces the size of feature maps by lowering their dimensions. In each dense block, dimensions of feature maps are constant, but number of filters changes. A transition layer is placed between each dense block to concatenate all previous inputs, reducing the number of channels and parameters needed in the network. Also, a bottleneck layer is placed between every layer to reduce the number of inputs, especially in far-away layers. DenseNet also introduced a growth rate parameter to regulate the quantity of information added in each layer. Most popular implementations are DenseNet121 and DenseNet201, where the number denotes the number of layers in the network.

MobileNet [73] is a small and efficient CNN designed for mobile computer vision tasks. It is built of layers of depthwise separable convolutions, composed of depth-wise and point-wise layers. MobileNet also introduced width multiplier and resolution multiplier hyperparameters. The width multiplier decreases the computational power needed during training, and the resolution multiplier decreases the resolution of the input image during training. The most popular versions of MobileNet are MobileNetV1 and MobileNetV2. In comparison with MobileNet, MobileNetV2 introduced inverted residual blocks and linear bottlenecks. Also, the Relu activation function was replaced by Relu6 (ReLu with saturation at value 6), which improved the model's accuracy.

3.3. Recurrent Neural Network

In our experiments, we applied two of the most popular recurrent neural network structures: LSTM and GRU. The RNN predicts following word in the sentence at every time step based on its predecessors. LSTM and GRU are types of RNNs specially designed to capture long-range dependencies in sequential data, such as time series or natural language.

Long-short-term memory (LSTM) [74] was designed for long-sequence problems and can predict the next word in the sequence based on its predecessors. Each LSTM unit consists of three gates that control and monitor the information flow in LSTM cells. The forgetting gate decides which information from the previous iteration will be stored in the cell state or is irrelevant and can be forgotten. In the input gate, the cell attempts to learn new information. It quantifies the relevance of the new input value of the cell and decides whether to process it. The output gate transfers the updated information from the current iteration to the next iteration. The state of the cell also contains the information along with a timestamp.

Gated Recurrent Unit (GRU) [75] were introduced as the more straightforward alternative to LSTM. All the information is passed and maintained through hidden states across all the time steps. Forget and input gates were replaced by one update gate, which controls the amount of information from the previous and current steps used. The reset gate controls how much information is discarded. Compared to LSTM, GRU has fewer parameters because the forget and output gates are merged into one update gate. Thanks to that, training is faster, and structure is more straightforward, especially for modeling an information flow within a Neural Network. On the other hand, LSTM proved to be a better solution for modeling long dependencies than GRU. However, the neural network's actual performance depends on the dataset's task and specification.

3.4. Word Embedding Models

Word embedding is a vector representation of words fed to the RNN part of our deep learning model that allows for the representation of words in the context. In our encoder-decoder merge captioning model, the *Word embedding component* creates input for the RNN that directly predicts words based on previous ones.

One-hot encoding is the most straightforward word-encoding technique, where each word/token is encoded to the binary vector representation. The method is based on the dictionary created for all unique tokens in the corpus. A fixed-length binary vector of the size of a dictionary represents each word. Each word is replaced by its index in this dictionary encoded as '1' in the embedding vector, while its other elements are set to '0'. It is a straightforward technique that captures various words but misses their semantic relation. Furthermore, fixed-length vectors are sparse, which could be more computationally efficient. More sophisticated approaches to word embedding have been proposed to get shorter vector representation that reflects relations between words.

The most common embedding systems used for natural language processing and image captioning are Glove, Word2Vec, and FastText. The Word2Vec [76] method simultaneously captures semantic relations between words. It is based on two techniques: continuous bag of words (CBOW) allows the prediction of words from the context word list vector and the continuous skip-gram model, a simple one-layer neural network that predicts context based on a given word. The most often used sizes of pre trained Word2Vec models are 100, 200, and 300.

FastText [77] comes from the Word2Vec model but analyzes words as n-grams. An algorithm is similar to the CBOW from Word2Vec but focuses on a hierarchical structure, representing a word in a dense form. Each n-gram is a vector, and the whole phrase is a sum of those vectors. Training is similar to the CBOW to achieve a word embedding vector. The size of the vectors in pretrained FastText models typically ranges from size 100 to 300. Models trained on large corpora have higher-dimensional vectors, whereas small corpora are represented in lower-dimensional ones.

Glove [78] word embedding is based on unsupervised learning to capture words that occur together frequently. Thanks to global and local statistics, semantic relations are created based on the whole corpus. Furthermore, it uses global matrix factorization to represent the word or lack of words in the document. It is also called the count-based model because Glove tries to learn how the words co-occur with other words in the corpus, allowing it to reflect the meaning of the words conditionally of the different words. Most popular pre-trained Glove vectors represent words in 200 or 300-dimensional space.

In our image captioning model, we utilized GLOVE and FastText word embedding, which were trained on a large corpus and allowed for the contextual modeling of words.

3.5. Adaptation, Merging and Word Prediction

The encoder-decoder merge model as input consumes images encoded by backbones (see section 3.2). *Adaptation component* is applied to transform features from the backbone into a format suitable for further processing. High-dimensional feature vectors are mapped into a lower-dimensional space, which helps to distill the most relevant visual information while reducing computational complexity, making it more manageable for subsequent stages of the captioning model.

Encoder and decoder are linked using merge architecture, meaning that features modified by the *Adaptation component* and output from the RNN are combined. In our experiments, we tested two ways of doing that: addition and concatenation.

In the add mechanism, modified image features and the RNN output are added element-wise – each corresponding element of the image features and the RNN output vectors are summed together. In this case, the adapted image feature vector and RNN output lengths must be equal.

The concatenate method creates a joint feature representation of modified image features and the RNN output by concatenating along the feature dimension axis. The length of the output vector is thus a sum of the lengths of the adapted image feature vector and the output of the RNN. The method combines both modalities and creates a comprehensive representation in a single feature vector.

Merged features are processed via a *Word prediction component* – a sequence of fully connected layers that first reduce the merged vector, transforming it into lower dimensional space. It is a trimmer to the final feature matrix that coherently and equally represents visual and text features. Finally, the probability distribution of the next word in the sequence is calculated using another fully connected layer. Output is thus a vector of size equal to the vocabulary size. A separate track of our experiments focused on the *Word prediction component* design.

3.6. Evaluation Metrics

Evaluating image captioning results is a complex task because it needs to determine if the machine's descriptions are correct and also judge their quality, relevance, and clarity, which can be subjective and complex.

Evaluation metrics in image captioning measure the correlation of generated captions with human judgment. The latter is present in the training and test set metadata in the form of 5 human-written captions (ground truth). Metrics are used to compare the model output with these five captions. They estimate the similarity between predicted captions and ground-truth ones. Evaluation metrics apply their technique for computation and have distinct advantages. Standard evaluation metrics for image captioning are BLEU-1 to BLEU-4, CIDEr, SPICE. They calculate word overlap between candidate and reference sentences. Higher values indicated better results.

BLEU (Bilingual Evaluation Understudy) [79] metric measures the correlation between predicted and human-made captions. It compares n-grams in predicted and reference sentences, where more common n-grams result in higher metric values. It is worth mentioning that metrics exclusively count n-grams; locations of the n-grams in sentences are not considered. Metric also allows addition weights for specific n-grams to prioritize longer, common sequences of words. Usually, 1 to 4 grams is used when computing the metric – the respective variants are called BLEU-1 up to BLEU-4.

CIDEr (Consensus-based Image Description Evaluation) [80] metric calculates correspondence between candidate and reference captions. It is based on the TF-IDF metric, calculated for each n-gram. It is widely used for SCST [34] training, where the strategy is to optimize the model for a specific metric. It results in higher results during the testing phase compared with [34]. Furthermore, CIDEr optimization during training impacts high BLEU, and SPICE metrics scores.

The Consensus-based Image Description Evaluation (CIDEr) [80] assesses how well a predicted caption aligns with the consensus among a set of reference image descriptions. It emphasizes similarity to human captions without requiring subjective judgments about factors such as content, grammar, or saliency.

All the above metrics are used in various NLP tasks. However, according to some investigations [81], they do not correlate with a human judgment, which makes them not adequate to measure the similarity of image captions [1]. Among the known metrics, the one that correlates with human judgment is SPICE (Semantic Propositional Image Caption Evaluation) [82]. This metric measures the similarity between sentences, represented by a directed graph. SPICE algorithm at the beginning creates two directed graphs. The first one is for all reference captions, and the second is for the candidate sentence. Graph elements can belong to three groups. The first group is objects and activity performers, the second group consists of descriptive tokens (adjectives adverbs), and the last group represents relations between objects and links other groups of tokens on the graph. Based on this representation, sentences are compared.

[1] The authors of [81] propose their metric, but due to much it's much lesser (more that 10x) popularity comparing with SPICE, we decided to use that latter in the current study.

4. Training and Evaluation

4.1. Dataset and Testing Procedure

During our experiments, we used for the evaluation and training of the MS COCO 2014 dataset [83, 84]. It consists of more than 120k images from various everyday scenes. Five captions describe each photo in natural language. We also used Karpathy split [18], where there are 113k images in training, 5k in validation, and 5k in test disjoint subsets. It is the most popular split in the image captioning area, allowing comparison models and results for them.

We divided our experiment into three stages to investigate each part of our encoder-decoder merge captioning model. Initially, we experimented with pre-trained image and text encoders to find the optimal pair for our model architecture. We trained various models using each image encoder and text encoder mentioned previously (see Table 1 for specific combinations of *Feature extraction and Word embedding* components). At that stage *Adaptation, RNN, Word prediction components* are excluded from the experimental process, and their parameters are configured permanently as not part of this experimental stage. Here, we look for the pair of *feature extraction and word embedding* components that produce the most accurate captions.

In the next step, we investigated the generalization of image and text features extracted via *Feature extraction and Word embedding* components. Here, responsible for that are *Adaptation, RNN, Word prediction, and MERGE* components. We set up permanent input and output features of *Feature extraction* and *word embedding* components and experimented with the parameters and structure of *Adaptation, RNN, Word prediction, MERGE components*. *Word prediction* component takes merged features from *MERGE componnent*, so we experimented respectively for add and concatenate method as merging strategy(see Tables 4 and 5), diving this area of experiments in two separate stages.

GRU is a popular alternative to LSTM in language modeling tasks, so we experimented with GRU as the basis of a language model. As in the previous scenario, we set up permanent input and output features of *Feature extraction and Word embedding* components and checked combinations of *Adaptation, RNN, Word prediction* components.

At each stage of our experimental setup, we evaluated results with BLEU-1 – BLEU-4, CIDEr, and SPICE metrics. The complete process will allow us to achieve a comprehensive perspective of the performance of different image encoders along with different embedding methods, separately to the influence of parameters in particular components of our image captioning model.

4.2. Training

Before training starts, captions are preprocessed – all words are converted to lowercase and tokenized. We removed punctuation, hanging single-letter words, and discarded rare words that occurred less than five times. Next, the list of processed words is appended with *start* and *stop* tokens to mark the beginning and end of the sentence, respectively. As a result, we achieved the MS COCO 2014 vocabulary (dictionary) that will be used to create an embedding matrix that contains embedding vectors. Before being handled by RNN, word sequences must be represented in word embedding vectors. We adopted pre-trained versions of FastText and Glove word embedding to extract the text features and finally achieved a vocabulary size of 7293. Each word is embedded into a 200-element vector for Glove and a 300-element vector for FastText word embedding space.

During training, the model processes combined RNN (LSTM or GRU) output and image feature vectors based on the backbone for a given image and *Adaptation component*. We applied different image features extractors described in section 3.2 and variants of the *Adaptation component, MERGE component, Word prediction component*, which are described in section 3.5. At each time step, the *Word prediction component* predicts a word for the processed image and compares it with the ground truth word from the corresponding training set. The predicted words and ground truth words (from the training set) are compared using the cross-entropy measure.

In our experiments, we us the Adam [85] optimizer with a base learning rate of *0.001*. We set the batch size to be *480* and train for up to *150* epochs with early stopping if the loss score had not improved over *0.001*.

Table 1. Evaluation results for MSCOCO 2014 test dataset. Metrics' values are averaged over the whole test dataset.

Image features	**No. of model parameters (mln)**	**Embeddings**	**Time of sentence generation (ms)**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**CIDEr**	**SPICE**
Vgg19	144.47	Glove	2046	64.1	45.83	31.86	22.34	69.62	13.93
	145.32	FastText	2090	65.42	46.89	32.72	22.93	71.79	14.46
Vgg16	143.26	Glove	2166	64.25	45.62	31.63	22.09	67.35	13.64
	144.11	FastText	2086	64.47	45.73	31.54	21.86	67.76	13.81
Resnet50	27.97	Glove	2468	65.33	47.26	33.26	23.44	73.12	14.43
	28.81	FastText	2016	65.97	47.82	33.79	24.02	74.47	14.71
Resnet152V2	62.71	Glove	2834	64.91	46.78	32.57	22.86	70.77	14.08
	63.55	FastText	2418	65.28	46.78	32.47	22.61	70.07	14.16
MobileNetV2	6.45	Glove	2096	65.39	47.14	33.04	23.24	73.03	14.55
	7.29	FastText	2144	65.13	47.22	33.17	23.32	73.79	14.62
MobileNet	8.36	Glove	3860	64.35	46.14	32.12	22.42	69.28	13.76
	9.2	FastText	1952	65.02	46.93	32.85	23.02	71.24	14.31
Xception	25.24	Glove	2414	66.59	48.63	34.34	24.33	78.13	15.16
	26.08	FastText	2052	67.01	48.8	34.45	24.3	77.64	15.18
InceptionV3	26.18	Glove	1960	66.12	47.72	33.35	23.38	74.16	14.72
	27.02	FastText	1922	66.15	47.87	33.57	23.63	75.04	14.83
DenseNet201	22.67	Glove	1828	66.35	48.41	34.26	24.18	76.54	14.96
	23.51	FastText	1748	66.59	48.73	34.57	24.55	76.74	14.83
DenseNet121	11.16	Glove	2468	65.03	47.02	32.96	23.26	71.94	14.13
	12.00	FastText	2360	65.39	47.09	32.89	23.09	72.36	14.25
Sugano [63]			-	71.4	50.5	35.2	24.5	63.8	-
Lebret [62]			-	73	50	34	23	-	-
Karpathy [18]			-	62.5	45	32.1	23	66	-
Xu [86]			-	67.9	49.3	34.7	24.3	75.4	-

4.3. Evaluation

During the testing, the image captioning model is fed by a preprocessed photo based on the backbone of a given image. In the beginning, at the 0-time step, there is no previously predicted word. Therefore, to denote the start of prediction, a start of sentence token *start* is used. Words are replaced by their embeddings. Next, the image captioning model predicts words recursively until the sentence's end (marked by *stop* token) or the maximum length of the sentence has been reached and adds it to the word list. At each step, the chance of the occurrence of one word next to another is calculated using embedding specific to the tested text features. Finally, a full caption for the tested image is generated and compared with ground-truth phrases for the tested image using specific metrics mentioned in 3.6.

During the evaluation, we used a greedy search algorithm when sampling the caption for the MS COCO 2014 dataset. We report results using the MS COCO captioning evaluation tool [84], which reports the following metrics: BLEU-1 – BLEU-4, CIDEr, and SPICE.

5. Results

5.1. Feature Extraction and Word Embedding

Motivated by [18], and considering the variety of available pre-trained object detection models and language processing models, at first, we conducted experiments to enhance the accuracy of a baseline captioning model via changes in encoding of an input data – the choice of the backbone in *Feature extraction component* with two types of word processing – the *Word embedding component*.

Images from the dataset are resized and normalized before entering the image captioning model to be compatible with one of the image feature extractors. For VGG16, VGG19, Resnet152V2, Resnet50, DenseNet121, DenseNet201, MobileNet, MobileNetV2 input shape is 224x224x3 and for InceptionV3, Xception it is equal to 299x299x3. As a result, we obtained feature vectors with the following sizes, corresponding to the preprocessed input image: 4096-element vector for VGG16, VGG19; 2048-element for InceptionV3, Xception, and Resnet152V2; 1024-element for DenseNet121; 1920-element for DenseNet201; 1000-element for MobileNet; and finally, 1280-element for MobileNetV2. We used backbones pre-trained on the ImageNet dataset, where the network's fully connected layers are removed since we do not need the probability distribution on 1000 image categories from ImageNet. The *Adaptation component* in this experiment consists of a single fully connected layer where the size of an input equals the length of a feature vector (which depends on the backbone used) and the output, the size of which equals the size of the RNN which is in this experiment, the LSTM network.

The *word prediction component* consists of two fully connected layers: the first keeps the same input and output size equal to 256, and the second outputs the probability of words. The size of the latter's input equals 256, while the output is the vocabulary size.

Table 1 shows the results of image captioning metrics calculated for different image extractors (column Image features) and text feature extractors (Embeddings column). We analyzed all models using the BLEU-1 – BLEU-4, CIDEr, and SPICE metrics. Following the literature, we used the most recent CIDEr and SPICE metrics to evaluate the performance, keeping the remainder for comparative purposes. We also added information about the number of parameters in the neural network(column "No. of model parameters (mln)") and the average time of sentence generation by the model(column "Time of sentence generation(ms)"). For the same purposes, we added four reference methods in the last four rows of Table 1.

From the obtained results, we can see that model performance depends mainly on the backbone used. Considering the CIDEr metric, the best results have been achieved for the Xception backbone feature extractor; second place belongs to DenseNet201. The spread between the highest (Xception with Glove, 78.13) and the lowest (VGG with Glove, 67.35) metrics value equals 10.78 points difference, which makes the model strongly dependent on the image backbone feature extractor. The evaluated quality of caption extractors is correlated with the backbones' accuracy. Average time of sentence generation is not correlated with the model complexity (no. of model params). Differences in execution time [2] spread from 1748 to 3860 ms. The fastest is DenseNet201, which is also the second-best model.

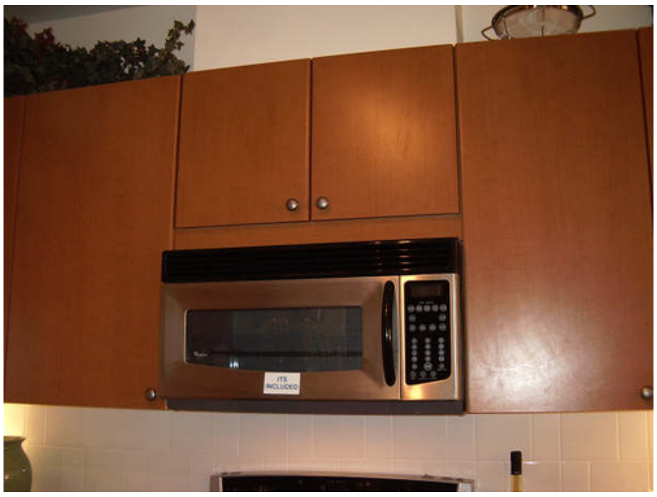
One cannot observe any remarkable superiority of one embedding model over another. For some metrics, the Glove model performs better, while for the remainder – the FastText. In most cases, FastText embedding achieves higher results than Glove for the same image features extractor. This suggests that FastText adapts more easily to different backbones than Glove. Longer feature vectors do not imply higher performance. The longest feature vectors VGG backbones generate do not mean higher measure values. The winning models use 2048- (Xception) and 1920-element (DenseNet201) vectors. The average time of sequence generation is not correlated with the model complexity (no. of model parameters). Differences in execution time between models spread from 874 to 1417 ms. The fastest is DenseNet201, which is also the second-best model.

In Table 2, we presented example correct captions obtained using various image feature extractors and Glove embedding; respective images are shown in Figure 2. The table contains ground-truth five captions from the dataset metadata, captions obtained from the model, and values of metrics calculated for particular predicted sentences. Generated captions sound good, are grammatically correct, and are consistent with the image content. However, on the picture 2b we can observe the hallucination phenomenon [87], despite no presence of the word "stove" in the ground truth captions

[2] All the execution times reported in this paper have been measured for a single caption on the computer with the following parameters: NVIDIA GeForce RTX 4070 with 12GB VRAM; AMD Ryzen 5 3600 6-Core Processor; 32GB RAM

model calculated that in the kitchen near the microwave must be a stove, even there is no such object on the image.

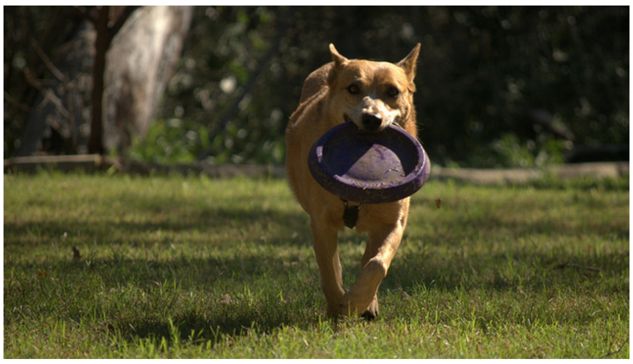
(a) ID = 95051



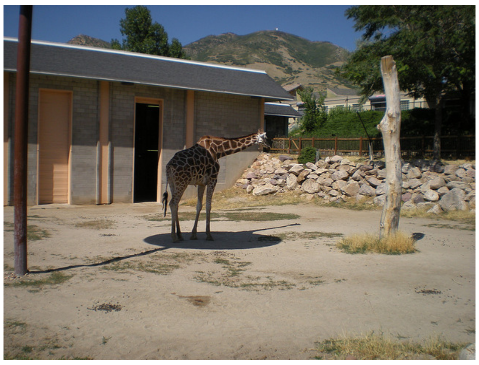
(b) ID = 309933

(c) ID = 570138

(d) ID = 105234



(e) ID = 347950



(f) ID = 27440

Figure 2. Images (with COCO ID's) with properly predicted captions (see Table 2 for details)

Table 2. Example images with predicted captions and achieved scores. For pictures see Figure 2

Image features	Image	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	Predicted caption	Ground truth captions
DenseNet121	2a	79.54	73.06	64.87	53.42	159.42	A man riding skis down a snow covered slope.	* A man on skis is posing on a ski slope. * A person on a ski mountain posing for the camera. * A man n a red coat stands on the snow on skis. * A man riding skis on top of a snow covered slope. * A lady is in her ski gear in the snow.
Resnet152V2	2e	60.00	44.72	0.00	0.00	83.88	A dog jumping in the air to catch a frisbee.	* A very cute brown dog with a disc in its mouth. * A dog running in the grass with a frisbee in his mouth. * A dog carrying a Frisbee in its mouth running on a grass lawn. * A dog in a grassy field carrying a frisbee. * A brown dog walking across a green field with a frisbee in it's mouth.
VGG19	2c	100.00	84.52	61.98	75.00	184.89	A bathroom with a toilet and a sink.	* A bathroom with a sink. toilet and vanity. * Tiled bathroom with a couple towels hanging up * An old bathroom with a black marble sink. * A bathroom with a black sink counter next to a white toilet. * The corner of a bathroom with light mint green walls above the tile.
MobileNet	2b	75.00	46.29	0.00	0.00	120.58	A kitchen with a stove and a microwave.	* A microwave is sitting idly in the kitchen. * A shiny silver metal microwave near wooden cabinets. * There are wooden cabinets that have a microwave attached at the bottom of it * A microwave sitting next to and underneath kitchen cupboards. * A kitchen scene with focus on a silver microwave.

In contrast to the above, in Table 3, we compared the caption for the picture 2d produced by the model, which achieved the overall highest CIDEr score in our research (Xception + Glove), with the model based on the DenseNet201. Presented metrics were explicitly calculated for predicted sentences. Comparing them, we can see that the model based on DenseNet201 does not correctly recognize objects in the image (bride and groom) and relations between them (cutting wedding cake). Human judgment is also reflected in the CIDEr metric. Despite grammatical correctness, the CIDEr metric rated the sentence 11.12 as unrelated to the five reference sentences. Result for the same sentence, produced by the model based on Xception, is 146.85 points higher, whereas in the produced sentence, we can point all of the relations and objects from reference sentences.

Table 3. Comparison of predicted captions for 2d, with DenseNet201 and Xception as image features extractors, along with Glove as word embeddings

Image	2d	
Image features	DenseNet201	Xception
BLEU-1	38.46	75.00
BLEU-2	17.90	65.47
BLEU-3	0.00	52.28
BLEU-4	0.00	41.11
METEOR	21.90	21.81
ROUGE-L	35.62	69.85
CIDEr	11.12	157.97
Predicted caption	A woman in a red dress is holding a white and red toothbrush.	A bride and groom cutting their wedding cake.
Ground truth captions	* A man and woman standing in front of a cake. * A newly wed couple celebrating with a toast. * A bride and groom celebrate over a cake. * A bride and groom are celebrating with wedding cake. * A man and a woman standing next to each other.	

5.2. Merging and Word Prediction

In the next step, we kept the best performing *Feature extraction* and *Word embedding components*, and we focused on her components that generalize image and text features in our image captioning model. *Adaptation component* as input consumes 2048 image features vector obtained via Xception backbone in the *Feature extraction component*. Images before entering the network are resized to 229x229x3 to be compatible with the Xception architecture. The *Adaptation component* reduces the length of a feature vector. Four cases are considered in our research: 128, 256, and 512. In addition, we also consider the case when the image feature vector is delivered directly to the *MERGE component* without any preprocessing.

The *Word embedding component* uses Glove embedding and, as input, consumes a vector of size 7293 – equal to the vocabulary size and output vector of size 200 – equal to the Glove embedding size. Next, the *RNN component* consumes embedded words and processes them through RNN, which in this experimental scenario is LSTM. The LSTM output size equals the size of the *Adaptation component's* output. However, in some of the experiments, we disregarded the *Adaptation component* (marked as "-" in the column "Adaptation component size" in the Table 5). In that case, concatenation directly merges Xception image features with LSTM output in *MERGE component*.

The *Word prediction component* comprises two fully connected layers. The first has input and output sizes equal to the output from *MERGE component*, while the second output is the probability of words vector – the vocabulary size. In some of the experiments, we disregarded the first fully connected layer(marked as "-" in the column "Word prediction component size" in the Tables 4 and 5) and directly sent merged image and text features to the layer that outputs the probability of words. This allowed us to test whether the output of the *MERGE component* needs to be trained in the *Word prediction component* to train the whole captioning model efficiently.

In the Tables 4 and 5, we show results of the experiments on the choice of parameters in *Adaptation*, *RNN, Word prediction components* for add and concatenate method, as the merging strategy in the *MERGE component*, respectively. Each row represents a variant of the model.

The first four rows of Tables 4 and 5 show comparable results for add and concatenate merging methods. Comparing them, we can see that using the concatenate merge method aggregates better local features into global features. We can achieve the same results as for the add-merge method but with fewer parameters in the network and significant reduction of training time. Best results considering the CIDEr metric 82.49 have been achieved for RNN and *Adaptation component* of size 256 (row 4 in Table 5), with *27.31 mln* parameters in the network. On the other hand, the model based on the add method (see Table 4) requires the size of RNN and *Adaptation component* equal to 512 units, with *28.82 mln* parameters in the network. Also comparing time of sentence generation, concatenate method performs faster. For our best results, concatenate method need average *2812 ms* and *5565 ms* for comparable add model.

In the row 9 in Table 5, we have a model trained without a fully connected layer after the merge component("Word prediction component size" column equal to "-"), comparable with row 1 from Table 4, where that layer was used. 3.19 difference in CIDEr metric clearly shows the importance of a fully connected layer that makes merged text and image features more understandable for the LSTM during the word prediction stage. Despite 3.47 more parameters and twice more time of sentence generation, method with a fully connected layer after the merge component achieved better results. Relu activation function present in the fully connected layer introduces non-linearity, which allows us to learn complex mapping between input features and output, which is the probability of words.

5.3. Recurrent Neural Network Model

In all experiments performed up to this moment, we used LSTM as the *RNN component* – part of the model that directly produces sentence word by word. Motivated by the [75], we reproduced a few experiments by learning a model with GRU as the RNN to check, to which extent the results depends of the type of RNN used in the captioning model.

We used a pre-trained Xception image features extractor with a 2048 output size in the *Feature extraction component*. As the *Word embedding component*, we utilized Glove, which serves a 200-element vector to the *RNN component* – here the GRU. We considered the concatenate method as a merging strategy in the *MERGE component*.

Comparing rows 4, 7, 8 from 5 and 1, 2, 3 from Table 6 respectively, one can observe that we achieved similar CIDEr results for LSTM and GRU, with the tendency of GRU to achieve higher CIDEr results in the models with fewer parameters. Also number of model parameters and time of sentence generation are similar for models based on LSTM and GRU. Comparing row 7 from Table 5 and row 2 from Table 6, we achieved results of 0.34 better in the GRU. Also, longer *Word prediction component* sizes do not imply higher measure values. Winning configuration utilizes a 256-word prediction component size, which means that GRUs perform better on less complex dependency tasks than LSTMs, which are fit for deep neural networks. From the design perspective GRUs perform well on short sequences, but in the image captioning domain we produce sequences with maximum of 50 tokens, which is short.

Table 4. Evaluation results for MSCOCO 2014 test dataset using LSTM as RNN and merging via add method.

	RNN size	Adaptation component size	Word prediction component size	No. of model parameters (mln)	Time of sentence generation (ms)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	SPICE
1	512	512	512	28.82	5565	67.53	49.67	35.39	25.19	82.48	15.75
2	256	256	-	25.18	4195	66.87	48.61	34.17	24.00	78.55	15.39
3	128	128	128	23.7	5100	65.86	47.94	33.54	23.40	74.92	14.77
4	256	256	256	25.2	6336	66.59	48.63	34.34	24.33	78.13	15.16

Table 5. Evaluation results for MSCOCO 2014 test dataset using LSTM as RNN and merging via concatenate method.

	RNN size	Adaptation component size	Word prediction component size	No. of model parameters (mln)	Time of sentence generation (ms)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	SPICE
1	512	512	1024	33.35	6024	66.31	48.47	34.29	24.22	79.05	15.57
2	256	256	-	27.04	2814	66.96	48.74	34.32	24.28	79.07	15.54
3	128	128	256	24.68	2647	67.27	49.69	35.36	25.04	80.44	15.71
4	**256**	**256**	**512**	**27.31**	**2812**	**67.51**	**49.75**	**35.56**	**25.36**	**82.49**	**16.08**
5	256	-	-	25.56	2765	65.36	47.02	32.72	22.77	75.93	14.90
6	256	-	512	26.66	2513	66.22	48.39	34.32	24.27	78.71	15.09
7	256	256	256	25.32	2247	67.56	49.72	35.48	25.24	81.85	15.63
8	256	256	128	24.31	1905	67.18	49.47	35.19	24.90	80.73	15.40
9	512	512	-	32.29	2393	65.24	47.33	33.30	23.54	75.86	14.88

Table 6. Evaluation results for MSCOCO 2014 test dataset using GRU as RNN.

	RNN size	Adaptation component size	Word prediction component size	No. of model parameters (mln)	Time of sentence generation (ms)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	CIDEr	SPICE
1	256	256	512	27.19	2211	67.32	49.62	35.51	25.45	81.41	15.60
2	256	256	256	25.19	2234	67.98	50.22	35.83	25.42	82.19	15.76
3	256	256	128	24.19	1823	67.20	49.10	34.67	24.41	80.39	15.33

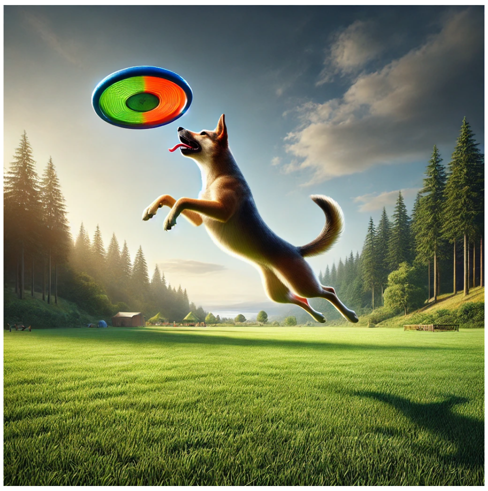
5.4. Experiments on External Images

Methods for generating image captions consist of multiple components that cooperate. Depending on their choice, models produce different captions, even for the same picture. For example, in the Table 7, we presented captions produced for the picture 2f. One may easily observe how the predicted caption differs from one model to another – each caption is different from the remainder, and so is the CIDEr measure. The overall quality of a captioning model is evaluated by measures (like CIDEr) averaged over the complete test dataset.

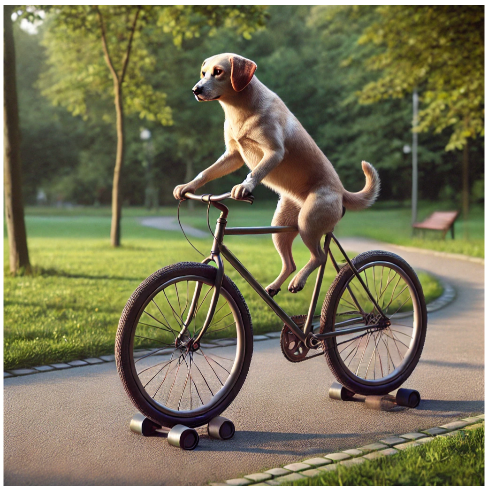
The image captioning models (as usual in machine learning) are intended to be applied to external images, i.e., images that are not present in the database that were used neither for training nor for validating the model. To investigate the ability of models to work correctly with such images, we also tested the best of our models on external data. This way, we validated if the model can generalize its expertise to unknown data. Our experiments revealed that the ability to generalize strongly depends on the image's content. In particular, if within the scene presented on an image are regions, objects similar to those included in some image in the training set, the generated caption is more correct. On the other hand, objects and scenes that have not been seen within the image in the training set are poorly described, primarily by irrational and incorrect captions.

For the image 3e model falsely recognized "sheep" as the object present on the scene. It is caused by the lack of "lama" object representation in the MSCOCO 2014 dataset; however, the surrounding scene is correct. For the Figure 3e predicted caption is totally pointless, "dolphins" are not present in the training dataset

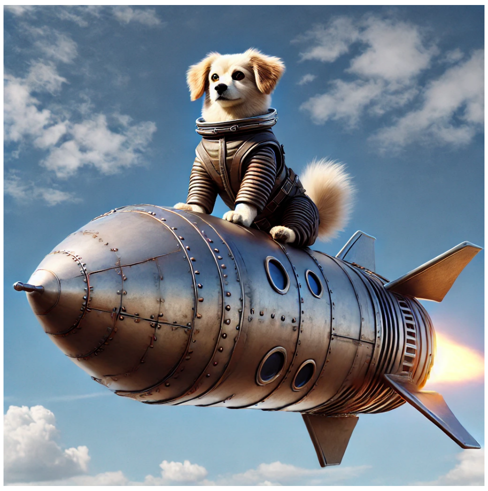
To further investigate the relation between the correctness of captions and the image content, artificial images were used. To generate them, the model DALL-E3 [88] in the ChatGPT4o [89] was used. As we may see, for the Figure 3a model correctly predicted caption and recognized objects present in the MSCOCO 2014 categories: frisbee and dog. On the other hand, for Figure 3c, Figure 3d where categories "rocket" and "tiger" are not included in MSCOCO 2014 categories, captions do not correctly describe images. Also, for Figure 3b, we can see that the category "dog" that was perfectly described in Figure 3b does not always guarantee the correct object category in the predicted caption. Object "dog" was mistaken with "man", but all relations (riding a bike) between other objects were perfectly recognized.



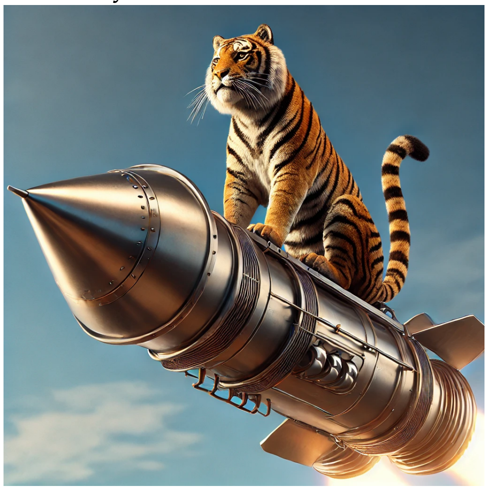
(a) "A dog jumping up to catch a frisbee."



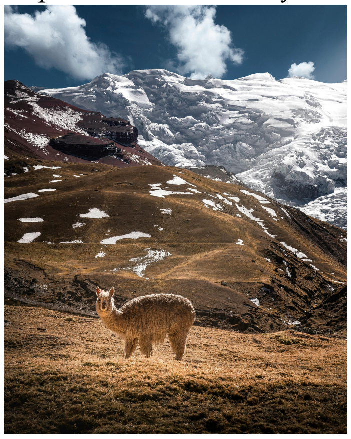
(b) "A man is riding a bike on a city street."



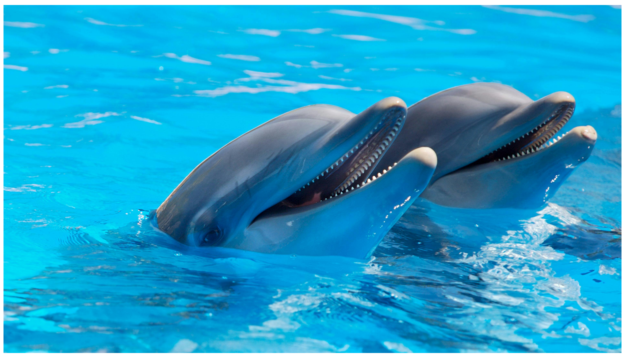
(c) "A large white and black airplane is on a runway."



(d) "A white and black cat is sitting on a red and black chair."



(e) "A sheep grazing on a mountain side by side."



(f) "A woman in a bikini is holding a surfboard."

Figure 3. Images with predicted captions generated artificially:(a) "Dog with frisbee", (b) "Dog on bike" , (c) "Dog on rocket" , (d) "Tiger on rocket"; real images with objects not present in the MS COCO dataset (e), (f).

Table 7. Example predicted captions and achieved scores for picture 2f

	CIDEr	Predicted caption
Image features: Xception; **merge method:** concatenate; **word prediction component:** 512; No adaptation component	2.2237	A giraffe standing in a field next to a tree.
Image features: VGG16; **merge method:** concatenate; **word prediction component:** 256;**RNN:** LSTM	1.7200	A giraffe standing next to a tree in a park.
Image features: Xception; **merge method:**concatenate; **word prediction component:** 256; **RNN:** GRU	1.6128	A giraffe standing in a dirt field next to a building.
Image features: MobileNetV2; **merge method:** concatenate; **word prediction component:** 256; **RNN:** LSTM	1.5162	A giraffe standing in a fenced in area.
Image features: Resnet50; **merge method:** concatenate; **word prediction component:** 256; **RNN:** LSTM	1.4194	A giraffe standing next to a zebra in a zoo.
Image features: Xception; **merge method:** concatenate; **word prediction component:** 512; **RNN:** GRU	1.2934	A giraffe standing next to a zebra in a field.
Image features: InceptionV3; **merge method:** concatenate; **word prediction component:** 256; **RNN:** LSTM	1.2851	A giraffe standing next to a wooden fence.
Image features: Xception; **merge method:** concatenate; **word prediction component:** 512; **RNN:** LSTM	0.9393	A couple of giraffe standing next to each other.
Ground truth captions		* A giraffe standing outside of a building next to a tree. * A giraffe standing in a small piece of shade. * A giraffe finds some sparse shade in his habitat. * Giraffe standing in a holding pen near a tree stump. * A giraffe in a zoo enclosure next to a barn.

5.5. Comparison with Transformer-Based Approaches

Despite promising results, most recent transformer-based models have certain disadvantages in image captioning. One of the primary issues is their high computational complexity and resource intensity [90]. The self-attention mechanism in transformers has a quadratic complexity concerning the sequence length, leading to substantial memory requirements for training, mainly when dealing with high-resolution images and long captions [45,91]. The training process is computationally expensive and time-consuming, necessitating powerful hardware such as GPUs or TPUs. Transformers also require large amounts of well-annotated data to learn effectively. Insufficient data can lead to overfitting and poor generalization, resulting in models that perform well on training data but poorly on unseen data [92].

Transformer models struggle with interpretability and explainability. Understanding and interpreting why a model generates a specific caption for an image can be challenging, which is a significant drawback in applications that can not work as "black-box" [93]. Integrating visual and text features in transformers is another complex task. While vision transformers (ViTs) have been developed, combining these with language models for image captioning remains less mature than traditional CNNs coupled with RNNs [94]. Furthermore, the performance of transformers can degrade with very long sequences, which can be an issue for complex and detailed image descriptions. Moreover, transformer architecture has not been initially designed for image processing tasks and requires substantial computational resources, significantly limiting their applicability.

Let us focus on examples to compare the results obtained with some transformer-based approaches on the COCO database. In the method [46], transformer architecture is used for both the encoder and decoder, eliminating convolutional layers. Authors achieved 129.4 CIDEr score, 81.7 BLEU-1, 40.0 BLEU-4, with 138.5 mln parameters in the network [90]. Solution [47] focused on focused on multi-modal reasoning. Achieved 132.8 CIDEr score, 80.9 BLEU-1, 39.7 BLEU-4, with 137.5 mln parameters in the network [90]. Model [50] focus on pretraining to capture more sophisticated relations between objects on images and achieved 80.9 BLEU-1, 39.5 BLEU-4, 129.3 CIDEr, with 138.2 mln parameters in the network [90]. In the model [52], authors improved the results of an existing image captioning model by developing a new visual features extractor. The new image captioning model consists of 369.6 mln parameters in the network [90] and obtained 82.0 BLEU-1, 41.0 BLEU-4, and 140.9 CIDEr.

On the other hand, our best model achieved 82.49 CIDEr metric, 67.51 BLEU-1, and 25.36 BLEU-4, with only 27.31 mln parameters in the network. One can see that the increase of the metrics value less than 1.6x has been obtained by the model of the complexity higher by one order of magnitude, i.e., almost 13x. Despite lower CIDEr results, sentences produced by our best model are grammatically and semantically correct while obtained using models of several times lower complexity (number of parameters).

6. Conclusions

Most recent transformer models are computationally intensive, requiring significant hardware and energy resources for training and inference. Classical models, which are the research subject, are more straightforward and, therefore, less resource-demanding. In the outcomes of our research, we showed that the substitution of the components of the classic model improves the overall efficiency. This paper analyzed how image features, word encoding, and manipulation of the generalization layers can enhance the encoder-decoder image captioning model results.

Experimenting with pre-trained image and text features extractors (*Word embedding component*, *Image features extractor*) proved that encoding input data plays the primary role in this area. During our research, we recognized that image captioning involves merging features from different modalities. Because of that, encoding images and features must cooperate, so finding the optimal pair for specific model architecture is crucial. We can significantly improve the results of the model predictions with that principle. The influence of the image feature extractor by the backbone is essential in this captioning model; it affects the performance more than the word embedding scheme. According to our experiment, the Xception with Glove and DenseNet201 with FastText are the best combinations of models' components.

Moreover, our investigation delved into the nuanced process of merging image and text modalities, recognizing its potential to streamline network parameters and, by extension, expedite the training time of the image captioning model. The strategic selection of the merging function emerged as a pivotal consideration, with our analysis conclusively demonstrating the superior efficiency of the "concatenate" method. This choice yielded comparable results to the "add" approach and exhibited a notable reduction in parameters within the whole neural network, further optimizing model training time and time of sentence generation.

During our exploration of various model components and methodologies, we encountered no limitations in the efficacy of the GRU in addressing the challenges posed by image captioning tasks. We achieved convergent CIDER results for LSTM and GRU, as RNNs, with the tendency of GRUs to perform better on less complex tasks. Also, the number of model parameters and time of sentence generation are similar for configurations comparable to our image captioning models. We observed that models leveraging GRU architecture generate grammatically coherent captions aligned with the content depicted in the images. It is worth further investigating GRUs as the language decoder in the image captioning models, especially for the models that must be easy to understand along with the results.

The image captioning model heavily relies on the quality and diversity of training data, which can lead to biases and poor generalization across diverse real-world scenarios. The model's adaptability to entirely new domains or significantly different image types without extensive retraining is limited. Additionally, the computational complexity involved restricts its real-time application and scalability. From this point of view, simpler models exhibit advantages over more sophisticated ones. As a result of our research, we showed that replacing their components enhances the quality of generating image captions. Our study's outcomes apply to all the research works that have led to the development of the optimal encoder-decoder image captioning model.

References

1. Ramachandram, D.; Taylor, G.W. Deep Multimodal Learning: A Survey on Recent Advances and Trends. *IEEE Signal Processing Magazine* **2017**, *34*, 96–108. doi:10.1109/MSP.2017.2738401.
2. Zhang, X.; He, S.; Song, X.; Lau, R.W.; Jiao, J.; Ye, Q. Image captioning via semantic element embedding. *Neurocomputing* **2020**, *395*, 212–221. doi:https://doi.org/10.1016/j.neucom.2018.02.112.
3. Janusz, A.; Kałuża, D.; Matraszek, M.; Łukasz Grad.; Świechowski, M.; Ślęzak, D. Learning multimodal entity representations and their ensembles, with applications in a data-driven advisory framework for video game players. *Information Sciences* **2022**, *617*, 193–210. doi:https://doi.org/10.1016/j.ins.2022.10.097.
4. Zhang, W.; Sugeno, M. A fuzzy approach to scene understanding. [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems, 1993, pp. 564–569 vol.1. doi:10.1109/FUZZY.1993.327529.
5. Iwanowski, M.; Bartosiewicz, M. Describing images using fuzzy mutual position matrix and saliency-based ordering of predicates. 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2021, pp. 1–8. doi:10.1109/FUZZ45933.2021.9494549.
6. Kuznetsova, P.; Ordonez, V.; Berg, A.; Berg, T.; Choi, Y. Collective Generation of Natural Image Descriptions. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Association for Computational Linguistics: Jeju Island, Korea, 2012; pp. 359–368.
7. Li, S.; Kulkarni, G.; Berg, T.L.; Berg, A.C.; Choi, Y. Composing Simple Image Descriptions using Web-scale N-grams. Proceedings of the Fifteenth Conference on Computational Natural Language Learning; Association for Computational Linguistics: Portland, Oregon, USA, 2011; pp. 220–228.
8. Mitchell, M.; Han, X.; Dodge, J.; Mensch, A.; Goyal, A.; Berg, A.; Yamaguchi, K.; Berg, T.; Stratos, K.; Daumé, H. Midge: Generating Image Descriptions from Computer Vision Detections. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics; Association for Computational Linguistics: USA, 2012; EACL '12, p. 747–756.
9. Farhadi, A.; Hejrati, M.; Sadeghi, M.A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; Forsyth, D. Every Picture Tells a Story: Generating Sentences from Images. Computer Vision – ECCV 2010; Daniilidis, K.; Maragos, P.; Paragios, N., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; pp. 15–29.
10. Barnard, K.; Duygulu, P.; Forsyth, D.; Blei, D.; Kandola, J.; Hofmann, T.; Poggio, T.; Shawe-Taylor, J. Matching Words and Pictures. *Journal of Machine Learning Research* **2003**, *3*. doi:10.1162/153244303322533214.
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems; Pereira, F.; Burges, C.; Bottou, L.; Weinberger, K., Eds. Curran Associates, Inc., 2012, Vol. 25.
12. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2015**, pp. 3156–3164.
13. Ramisa, A.; Yan, F.; Moreno-Noguer, F.; Mikolajczyk, K. BreakingNews: Article Annotation by Image and Text Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2018**, *40*, 1072–1085.
14. Biten, A.F.; Gómez, L.; Rusiñol, M.; Karatzas, D. Good News, Everyone! Context Driven Entity-Aware Captioning for News Images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2019**, pp. 12458–12467.
15. Sharma, P.; Ding, N.; Goodman, S.; Soricut, R. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. ACL, 2018.
16. Changpinyo, S.; Sharma, P.; Ding, N.; Soricut, R. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. CVPR, 2021.
17. Kiros, R.; Salakhutdinov, R.; Zemel, R. Multimodal Neural Language Models. Proceedings of the 31st International Conference on Machine Learning; Xing, E.P.; Jebara, T., Eds.; PMLR: Bejing, China, 2014; Number 2 in Proceedings of Machine Learning Research, pp. 595–603.
18. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3128–3137. doi:10.1109/CVPR.2015.7298932.
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. doi:10.1109/CVPR.2014.81.

20. Donahue, J.; Hendricks, L.A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Darrell, T. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 677–691. doi:10.1109/TPAMI.2016.2599174.
21. Johnson, J.; Karpathy, A.; Fei-Fei, L. Densecap: Fully convolutional localization networks for dense captioning. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4565–4574.
22. Xiao, X.; Wang, L.; Ding, K.; Xiang, S.; Pan, C. Dense semantic embedding network for image captioning. *Pattern Recognition* **2019**, *90*, 285–296. doi:https://doi.org/10.1016/j.patcog.2019.01.028.
23. Toshevska, M.; Stojanovska, F.; Zdravevski, E.; Lameski, P.; Gievska, S. Exploration into Deep Learning Text Generation Architectures for Dense Image Captioning. 2020 15th Conference on Computer Science and Information Systems (FedCSIS), 2020, pp. 129–136. doi:10.15439/2020F57.
24. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. Proceedings of the 32nd International Conference on Machine Learning; Bach, F.; Blei, D., Eds.; PMLR: Lille, France, 2015; Vol. 37, *Proceedings of Machine Learning Research*, pp. 2048–2057.
25. Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; Zhang, L. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* **2018**, pp. 6077–6086.
26. Guo, L.; Liu, J.; Tang, J.; Li, J.; Luo, W.; Lu, H. Aligning Linguistic Words and Visual Semantic Units for Image Captioning. Proceedings of the 27th ACM International Conference on Multimedia; Association for Computing Machinery: New York, NY, USA, 2019; MM '19, p. 765–773. doi:10.1145/3343031.3350943.
27. Gu, J.; Wang, G.; Cai, J.; Chen, T. An Empirical Study of Language CNN for Image Captioning. *2017 IEEE International Conference on Computer Vision (ICCV)* **2016**, pp. 1231–1240.
28. Liu, S.; Bai, L.; Hu, Y.; Wang, H. Image Captioning Based on Deep Neural Networks. *MATEC Web of Conferences* **2018**, *232*, 01052. doi:10.1051/matecconf/201823201052.
29. Subash, R.; Jebakumar, R.; Kamdar, Y.; Bhatt, N. Automatic Image Captioning Using Convolution Neural Networks and LSTM. *Journal of Physics: Conference Series* **2019**, *1362*, 012096. doi:10.1088/1742-6596/1362/1/012096.
30. Xu, K.; Wang, H.; Tang, P. Image captioning with deep LSTM based on sequential residual. 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 361–366. doi:10.1109/ICME.2017.8019408.
31. Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Yuille, A.L. Explain Images with Multimodal Recurrent Neural Networks. *CoRR* **2014**, *abs/1410.1090*, [1410.1090].
32. Dong, H.; Zhang, J.; McIlwraith, D.; Guo, Y. I2T2I: Learning Text to Image Synthesis with Textual Data Augmentation. 2017 IEEE International Conference on Image Processing (ICIP). IEEE Press, 2017, p. 2015–2019. doi:10.1109/ICIP.2017.8296635.
33. Xian, Y.; Tian, Y. Self-Guiding Multimodal LSTM-When We Do Not Have a Perfect Training Dataset for Image Captioning. *IEEE Transactions on Image Processing* **2017**, *PP*. doi:10.1109/TIP.2019.2917229.
34. Rennie, S.J.; Marcheret, E.; Mroueh, Y.; Ross, J.; Goel, V. Self-Critical Sequence Training for Image Captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2017**, pp. 1179–1195.
35. Lu, J.; Xiong, C.; Parikh, D.; Socher, R. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2017**, pp. 3242–3250.
36. Delbrouck, J.; Dupont, S. Bringing back simplicity and lightliness into neural image captioning. *CoRR* **2018**, *abs/1810.06245*, [1810.06245].
37. Tanti, M.; Gatt, A.; Camilleri, K. What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator? Proceedings of the 10th International Conference on Natural Language Generation; Alonso, J.M.; Bugarín, A.; Reiter, E., Eds.; Association for Computational Linguistics: Santiago de Compostela, Spain, 2017; pp. 51–60. doi:10.18653/v1/W17-3506.
38. Zhou, L.; Xu, C.; Koch, P.A.; Corso, J.J. Image Caption Generation with Text-Conditional Semantic Attention. *ArXiv* **2016**, *abs/1606.04621*.
39. Chen, X.; Zitnick, C.L. Mind's eye: A recurrent visual representation for image caption generation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2422–2431. doi:10.1109/CVPR.2015.7298856.

40. Hessel, J.; Savva, N.; Wilber, M. Image Representations and New Domains in Neural Image Captioning **2015**. doi:10.18653/v1/W15-2807.
41. Song, M.; Yoo, C.D. Multimodal representation: Kneser-ney smoothing/skip-gram based neural language model. 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 2281–2285. doi:10.1109/ICIP.2016.7532765.
42. Hendricks, L.; Venugopalan, S.; Rohrbach, M.; Mooney, R.; Saenko, K.; Darrell, T. Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE Computer Society: Los Alamitos, CA, USA, 2016; pp. 1–10. doi:10.1109/CVPR.2016.8.
43. You, Q.; Jin, H.; Wang, Z.; Fang, C.; Luo, J. Image Captioning with Semantic Attention. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE Computer Society: Los Alamitos, CA, USA, 2016; pp. 4651–4659. doi:10.1109/CVPR.2016.503.
44. Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Yuille, A.L. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *arXiv: Computer Vision and Pattern Recognition* **2014**.
45. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv* **2020**, *abs/2010.11929*.
46. Wei Liu, Sihan Chen, L.G.X.Z.J.L. CPTR: FULL TRANSFORMER NETWORK FOR IMAGE CAPTIONING, 2021, [arXiv:cs.CV/2101.10804].
47. Pan, Y.; Yao, T.; Li, Y.; Mei, T. X-linear attention networks for image captioning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10971–10980.
48. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. International Conference on Machine Learning, 2021.
49. Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.; Parekh, Z.; Pham, H.; Le, Q.V.; Sung, Y.; Li, Z.; Duerig, T. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. *CoRR* **2021**, *abs/2102.05918*, [2102.05918].
50. Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J.; Gao, J. Unified vision-language pre-training for image captioning and vqa. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 13041–13049.
51. Li, X.; Yin, X.; Li, C.; Zhang, P.; Hu, X.; Zhang, L.; Wang, L.; Hu, H.; Dong, L.; Wei, F.; others. Oscar: Object-semantics aligned pre-training for vision-language tasks. European Conference on Computer Vision. Springer, 2020, pp. 121–137.
52. Zhang, P.; Li, X.; Hu, X.; Yang, J.; Zhang, L.; Wang, L.; Choi, Y.; Gao, J. VinVL: Making Visual Representations Matter in Vision-Language Models. *CVPR 2021* **2021**.
53. Ding, Z.; Sun, Y.; Xu, S.; Pan, Y.; Peng, Y.; Mao, Z. Recent Advances and Perspectives in Deep Learning Techniques for 3D Point Cloud Data Processing. *Robotics* **2023**, *12*. doi:10.3390/robotics12040100.
54. Zhang, H.; Wang, C.; Yu, L.; Tian, S.; Ning, X.; Rodrigues, J. PointGT: A Method for Point-Cloud Classification and Segmentation Based on Local Geometric Transformation. *IEEE Transactions on Multimedia* **2024**, pp. 1–12. doi:10.1109/TMM.2024.3374580.
55. Wang, C.; Ning, X.; Sun, L.; Zhang, L.; Li, W.; Bai, X. Learning Discriminative Features by Covering Local Geometric Space for Point Cloud Analysis. *IEEE Transactions on Geoscience and Remote Sensing* **2022**, *60*, 1–15. doi:10.1109/TGRS.2022.3170493.
56. Wang, C.; Ning, X.; Li, W.; Bai, X.; Gao, X. 3D Person Re-Identification Based on Global Semantic Guidance and Local Feature Aggregation. *IEEE Transactions on Circuits and Systems for Video Technology* **2024**, *34*, 4698–4712. doi:10.1109/TCSVT.2023.3328712.
57. Xue, L.; Yu, N.; Zhang, S.; Panagopoulou, A.; Li, J.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J.C.; Savarese, S. ULIP-2: Towards Scalable Multimodal Pre-training for 3D Understanding. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 27091–27101.
58. Chen, G.; Wang, M.; Yang, Y.; Yu, K.; Yuan, L.; Yue, Y. PointGPT: Auto-regressively Generative Pre-training from Point Clouds. Thirty-seventh Conference on Neural Information Processing Systems, 2023.
59. Wang, S.S.; Dong, R.Y. Learning Complex Spatial Relation Model from Spatial Data. *Journal of Computers* **2019**, *30*, 123–136.

60. Yang, Z.; Zhang, Y.; ur Rehman, S.; Huang, Y. Image Captioning with Object Detection and Localization. *CoRR* **2017**, *abs/1706.02430*, [1706.02430].
61. Herdade, S.; Kappeler, A.; Boakye, K.; Soares, J. Image Captioning: Transforming Objects into Words. *CoRR* **2019**, *abs/1906.05963*, [1906.05963].
62. Lebret, R.; Pinheiro, P.O.; Collobert, R. Phrase-Based Image Captioning. Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. JMLR.org, 2015, ICML'15, p. 2085–2094.
63. Sugano, Y.; Bulling, A. Seeing with Humans: Gaze-Assisted Neural Image Captioning. *ArXiv* **2016**, *abs/1608.05203*.
64. Li, Y. Image Caption using VGG model and LSTM. *Applied and Computational Engineering* **2024**, *48*, 68–77. doi:10.54254/2755-2721/48/20241175.
65. Bartosiewicz, M.; Iwanowski, M.; Wiszniewska, M.; Frączak, K.; Leśnowolski, P. On Combining Image Features and Word Embeddings for Image Captioning. 2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS), 2023, pp. 355–365. doi:10.15439/2023F997.
66. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings; Bengio, Y.; LeCun, Y., Eds., 2015.
67. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.C.; Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **2015**, *115*, 211–252. doi:10.1007/s11263-015-0816-y.
68. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.
69. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9. doi:10.1109/CVPR.2015.7298594.
70. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2016**, pp. 2818–2826.
71. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *CoRR* **2016**, *abs/1610.02357*, [1610.02357].
72. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
73. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* **2017**, *abs/1704.04861*, [1704.04861].
74. Hochreiter, S.; Schmidhuber, J. LSTM Long Short-term Memory. *Neural computation* **1997**, *9*, 1735–80. doi:10.1162/neco.1997.9.8.1735.
75. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1724–1734. doi:10.3115/v1/D14-1179.
76. Mikolov, T.; Chen, K.; Corrado, G.S.; Dean, J. Efficient Estimation of Word Representations in Vector Space. International Conference on Learning Representations, 2013.
77. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* **2017**, *5*, 135–146.
78. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
79. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics; Association for Computational Linguistics: USA, 2002; ACL '02, p. 311–318. doi:10.3115/1073083.1073135.
80. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-based image description evaluation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4566–4575. doi:10.1109/CVPR.2015.7299087.

81. Cui, Y.; Yang, G.; Veit, A.; Huang, X.; Belongie, S. Learning to Evaluate Image Captioning. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE Computer Society: Los Alamitos, CA, USA, 2018; pp. 5804–5812. doi:10.1109/CVPR.2018.00608.
82. Anderson, P.; Fernando, B.; Johnson, M.; Gould, S. Spice: Semantic propositional image caption evaluation. European conference on computer vision. Springer, 2016, pp. 382–398.
83. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. Computer Vision – ECCV 2014; Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds.; Springer International Publishing: Cham, 2014; pp. 740–755.
84. Chen, X.; Fang, H.; Lin, T.; Vedantam, R.; Gupta, S.; Dollár, P.; Zitnick, C.L. Microsoft COCO Captions: Data Collection and Evaluation Server. *CoRR* **2015**, *abs/1504.00325*, [1504.00325].
85. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* **2014**.
86. Xu, N.; Liu, A.; Liu, J.; Nie, W.; Su, Y. Scene graph captioner: Image captioning based on structural visual representation. *J. Vis. Commun. Image Represent.* **2019**, *58*, 477–485.
87. Rohrbach, A.; Hendricks, L.A.; Burns, K.; Darrell, T.; Saenko, K. Object Hallucination in Image Captioning. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 4035–4045. doi:10.18653/v1/D18-1437.
88. OpenAI. DALL·E 3 System Card, 2023. Accessed: 2024-07-12.
89. OpenAI. Introducing GPT-4o and More Tools to ChatGPT Free Users, 2024. Accessed: 2024-07-12.
90. Stefanini, M.; Cornia, M.; Baraldi, L.; Cascianelli, S.; Fiameni, G.; Cucchiara, R. From Show to Tell: A Survey on Image Captioning. *CoRR* **2021**, *abs/2107.06912*, [2107.06912].
91. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. Advances in Neural Information Processing Systems; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Eds. Curran Associates, Inc., 2017, Vol. 30.
92. Radford, A.; Narasimhan, K. Improving Language Understanding by Generative Pre-Training **2018**.
93. Wiegreffe, S.; Pinter, Y. Attention is not not Explanation. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); Inui, K.; Jiang, J.; Ng, V.; Wan, X., Eds.; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 11–20. doi:10.18653/v1/D19-1002.
94. Tan, H.; Bansal, M. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); Inui, K.; Jiang, J.; Ng, V.; Wan, X., Eds.; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 5100–5111. doi:10.18653/v1/D19-1514.