DOI: 10.15439/2023F997
ISSN 2300-5963 ACSIS, Vol. 35

On combining image features and word embeddings for image captioning

Mateusz Bartosiewicz, Marcin Iwanowski, Martika Wiszniewska, Karolina Frączak, Paweł Leśnowolski
Institute of Control and Industrial Electronics, Warsaw University of Technology
ul. Koszykowa 75; 00-662 Warszawa POLAND
mateusz.bartosiewicz.dokt@pw.edu.pl; marcin.iwanowski@pw.edu.pl

***Abstract*—Image captioning is the task of generating semantically and grammatically correct caption for a given image. Captioning model usually has an encoder-decoder structure where encoded image is decoded as list of words being a consecutive elements of the descriptive sentence. In this work, we investigate how encoding of the input image and way of coding words affects the result of the training of the encoder-decoder captioning model. We performed experiments with image encoding using 10 all-purpose popular backbones and 2 types of word embeddings. We compared those models using most popular image captioning evaluation metrics. Our research shows that the model's performance highly depends on the optimal combination of the neural image feature extractor and language processing model. The outcome of our research are applicable in all the research works that lead to the developing the optimal encoder-decoder image captioning model.**

***Index Terms*—image captioning, neural image feature extractors, embedding models, LSTM**

I. Introduction

IMAGE captioning is a task of generating a verbal description of an image. It combines Natural Language Processing (NLP) and Computer Vision. Image captioning solutions are used in many application areas. They are adapted for content-based image retrieval or automated labeling of online images. Also, in the human-machine interaction field, they are used to assist visually impaired people in understanding the surrounding world or to search fast for photos on the internet.

We focus in this paper on the baseline captioning model [24] consisting of encoder and decoder. Encoder extracts a pair of image and text features in parallel. Text features encoder is responsible for the dense representation of each word in embedding space providing semantic context for each token. Image encoder uses convolutional neural network (CNN) backbone which extracts high-level image features. Decoder combines image and text features and generates the resulting image caption. It is based on the long-short term memory (LSTM) module [18], that generates the descriptive sentence word-by-word.

In this work, we improve the effectiveness of the baseline image captioning model by changing the encoding of the input data. We assume that different image features extractors, even pretrained on the same training set, provide with various high-level knowledge of the image content and similarly, different language processing models extract different semantics of captions.

During experiments, we investigated how different encoding of an image and text influence the captioning accuracy. We tested several backbone models based on pretrained CNN networks and embedding schemes as image and language inputs, respectively. It allowed us to investigate which pairs work best, hence finding the optimal combination of neural image feature extractor and language processing model.

As a result, we achieved 20 models trained on CNN networks: Xception, InceptionV3, Resnet152V2, Resnet50, VGG16, VGG19, DenseNet121, DenseNet201, MobileNet, MobileNetV2 along with Glove and FastText embeddings. For training and testing we used MSCOCO 2014 dataset and as the evaluation metrics: BLEU, METEOR, CIDEr, SPICE, ROGUE-L, WMD. Finally, thanks to the mentioned metrics, we assessed which pairs of image features and embeddings produce better results on the baseline image captioning model.

This paper is organized as follows. Section II describes how image captioning methods evolved from template-based techniques to deep neural architectures. Next, in section III, we describe how our base image captioning model is built and what neural image features extractors and language embedding models we use. The experimental procedure applied in our research is presented in Section IV. Section V have experimental analysis and finally, the final conclusions are found in Section VI.

II. Previous works

Image captioning methods combining text and visual data belong to the multi-modal machine-learning approaches [22], [40], [59]. Captioning models can be divided into traditional and deep-learning-based. Originally, traditional image captioning methods were based on hard-coded rules and human-made features. In [27], [29], [36], 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 [12], authors used already existing, predefined sentences. They created space of meaning from images features and compared images with sentences to find the most appropriate sentences for a photo. Despite semantic and grammatical correctness, captions from traditional methods differ often from the way a human described the image content.

Deep learning image captioning methods tries to overcome those limitations. In pioneering work [25] authors suggested that neural networks can interpret deep semantics of images and word embeddings. They proved that combined image features extracted by the convolutional neural networks (CNN) and word embeddings could hold semantic meaning. In [11], authors suggested passing image features and text features sequentially and individually to the language model. Inspired by the success in machine translation, [51] proposed using an encoder-decoder framework in image captioning, which has recently become dominant in the image captioning field.

Paper [24] by Karpathy et al. introduced architecture similar to human perception. Method generates novel descriptions over image regions, with R-CNN (Regional Convolutional Neural Networks) [13] for image feature extraction and recurrent neural network (RNN) to iteratively generate consecutive words of caption. Model using the multimodal embeddings space tries to find the parts of the sentence that best fits the image regions. Differently from other proposed methods ([9], [25], [51]), 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, focusing on generating captions for each region of an image, are called dense image captioning [23], [49], [54].

Encoder-decoder architecture [2], [15], [51], [55] considers the task of image captioning as the sequence-to-sequence problem. Encoder encodes the image to the fixed length vector using the image features extractor. Most widely used are CNN networks as VGG [14], [32], [45], ResNet [32], [34], [56] or Inception [10], [53]. Decoder, which in image captioning is represented by a language model, generates natural language descriptions to the output. Most popular approaches used RNN. However, due to the vanishing gradient problem that occurs in long sequence tasks, LSTM which is a variation of RNN achieved better results [18]. Most popular encoder-decoder approaches are the CNN-RNN [33], [41], [51] and GRU [8].

During the rapid development of image captioning methods, researchers also investigated other aspects of captions than just comparability to human judgment. Researchers focused on captions with a specific style. In [2], authors improved the descriptiveness of generated captions by combining CNN and LSTM. In [52], authors focused on captions for visually impaired people. Developed model tends to create captions that describe the surrounding environment.

III. Preliminaries

A. Image captioning model

Image captioning encoder-decoder model investigated in this study is depicted in Fig. 1. Encoder consists of two parts working, in the learning phase, simultaneously. One is for handling image features and another is for handling words in sequences. Firstly, image features are extracted using one of the image features extractors described in the next section. They are processed by a dense (fully connected layer) layer with ReLU (rectified linear unit) activation functions [37]. Its usage was motivated by promising results in very deep vision neural networks [17]. Compared with non-linear functions like sigmoid, ReLU is faster and harder to overfit. Dense layer is responsible for reducing the dimension of the image feature space (i.e. the length of the feature vector) to 256 to match the size of the word sequence prediction output.

In parallel, the text input (caption) is transformed into the sequence of indices of consecutive sentence words. Although the length of a caption varies, the length of vector of indices is constant and equal to 51, which is the maximum sentence lenght (i.e. number of words in the longest caption sentence). Such a vector is fed to the embedding layer. It encodes the semantic meaning of words represented by vectors in embedding space. We used pretrained Glove and FastText embeddings as two alternative ways of encoding the consecutive words of a descriptive sentence. Thanks to the embeddings layer, we reduced the text features size from the vocabulary size to the vectors of embeddings. Embedding vectors are passed through a long-short term memory (LSTM) model of size 256. After the LSTM layer, the outputs of language model and the image part of the image captioning model are added and finally forwarded to the decoder consisting of two dense layers.

Long-short term memory (LSTM) was designed for long-sequence problems and can predict 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. Forgetting gate decides, which information from 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 relevance of new input value of the cell and decides to process it or not. Output gate transfers the updated information from the current iteration to the next iteration. State of the cell also contains the information along with a timestamp.

Decoder processes an image feature vector and a sequence vector to predict captions. Following two dense layers processes, added language and image model to reduce the number of features to the vector of size equal to vocabulary size. Finally, the softmax layer generates the probability distribution of the next word in the sequence and selects the word with maximum probability. Previous words are converted to embeddings during training to develop the next word. Image feature vector is fed to the decoder. Goal of the training is to minimize loss function based on the error between target and predicted words.

Trained model predicts captions word-by-word, where the prediction of the next word is based on the previously generated one and image features. At each iteration, greedy search algorithm looks for the word in the dictionary with the highest probability of following words in the sequence. Process continues till the end of the caption is detected or the max length of the caption is achieved. Greedy search takes only tokens with the highest possibility of occurring in the final sequence based on previously generated tokens.

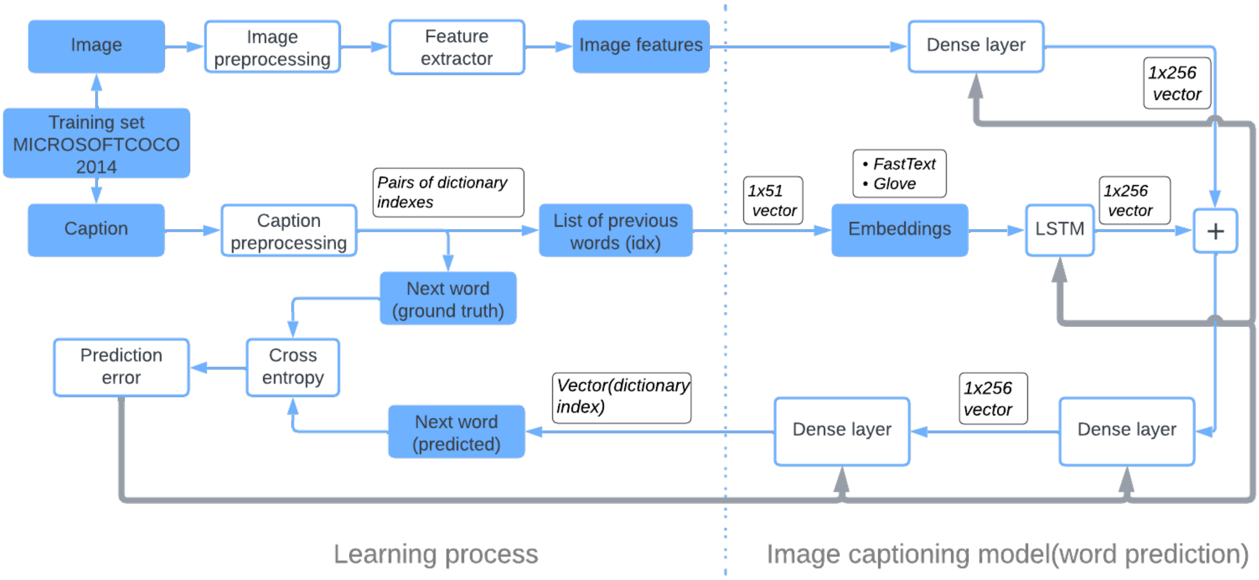


Fig. 1: Diagram of image captioning model training process.

B. *Neural image feature extractors*

Image features are essential in image captioning. In our experiments we used backbone CNN networks pretrained on a large number of images, the backbone networks. It makes possible to focus on the captioning model and restrict training to the remainder of the model.

The VGG [44] is a group of convolutional neural networks (CNNs) widely used for image classification tasks. Most popular variants are VGG16 and VGG19. VGG16 consists of 13 convolutional and 3 dense layers and was trained to recognize 1000 object classes referring to objects depicted on input 224x224x3 color images. By cutting out the dense layers, the backbone network that produces the image feature vector of length 4096 has been obtained. VGG19 has 3 more CNN layers than VGG16. Thanks to this, allows to learn richer representations of the data and achieves 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 [16] network was created to support many layers while preventing the phenomenon of vanishing gradient in deep neural networks. Most popular variants are Resnet18, Resnet50, and Resnet100, where the number represents a number of layers. Network architecture is built among two stages. In the beginning, the stack of skip connections 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 [47] model was created to deal with overfitting in very deep neural networks by going wider in layers rather than deeper. It is build among inception blocks that process input and repetitively passes the result to another inception block. Each block consists of four parallel layers 1x1, 3x3, 5x5, and max-pooling. 1x1 is to reduce dimension by channel-wise pooling. Thanks to that 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. Most popular versions of the Inception network are Inception, InceptionV2 and InceptionV3.

The InceptionV3 [48] incorporated the best techniques to optimize and reduce the computational power needed for images features extraction in the network. It is a deeper network than InceptionV2 and Inception, but its effectiveness was not compromised. Also, use auxiliary classifiers that 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 to fasten computations.

The Xception [6] is a variation of an Inception [47] model that decouples cross-channel correlations 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 module. It has a simple and modular architecture and achieved better results than VGG16, Resnet and InceptionV3 in classical classification challenges.

The backbone networks based on the three above ones, in contrast to the VGG16, produce the image feature vector of length 2048.

DenseNet [21] Network was created to overcome 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 allow to reuse parameters within network and produce models with high accuracy. Structure of DenseNet is based on stack of connectivity, transition and bottleneck layers, grouped in dense blocks. Every layer is connected, with every another layer in dense way. Dense block is 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 change. Between each dense block, transition layer is placed to concatenate all previous inputs, hence reduce number of channels and number of parameters needed in the network. Also, between every layer bottleneck layer is placed to reduce number of inputs especially in far away layers. DenseNet also introduced growth rate parameter to regulate quantity of information added in each layer. Most popular implementations are DenseNet121, DenseNet201, where number denotes quantity of layers in the network.

MobileNet [20] is a small and efficient CNN Network especially 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. Width multiplier allows to decrease computational power needed during training, resolution multiplier decreases the resolution of the input image during training. Most popular versions of MobileNet are MobileNetV1 and MobileNetV2. In comparison with MobileNet, MobileNetV2 introduced inverted residual blocks and linear bottlenecks. Also, Relu activation function was replaced by Relu6 (ReLu with saturation at value 6). Thanks to that accuracy of the model significantly improved.

C. *Word embedding models*

Word embeddings are vector representations of tokens that are fed to a deep learning model. The most common embedding systems used for natural language processing and image captioning are Glove, Word2Vec and FastText.

One of the first word embedding techniques was one-hot encoding, where each token is encoded to the binary vector representation. Method is based on the dictionary created for all unique tokens in the corpus. A fixed-length binary vector with the size of a dictionary represents each word. Index of the word in the vector represents presence. If a word is present in with vector, just one value is one and others are 0. It is a straightforward technique that captures a wide variety of words but misses the semantic relation of words. Furthermore, fixed-length vectors are sparse, which is not computationally efficient.

Computationally efficient, Word2Vec [35] method simultaneously captures semantic relations between words. It is based on two techniques: CBOW (Continuous Bag of Words) 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.

FastText [4] 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. To achieve a word embeddings vector, training is similar to the CBOW.

Glove [39] word embeddings are based on unsupervised learning to capture words that occur together frequently. Thanks to the global and local statistics, it creates semantic relations in the whole corpus. Furthermore, it uses global matrix factorization to represent the word of 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 other words.

D. *Text evaluation metrics*

Image captioning is a task that belongs to both computer vision and natural language processing (NLP) domains. It must capture objects, the relations between them and the whole scene context to produce readable sentences in natural language. Due to the complexity of the image captioning results, the evaluation of the image captioning is still a complicated and comprehensive problem.

Evaluation metrics in image captioning measure the correlation of generated captions with human judgment. They estimate grammatical correctness, the complexity of the description and how generated caption generalizes the corresponding image. Evaluation metrics apply their own technique for computation and have distinct advantages. Standard evaluation metrics for image captioning are BLEU-1 to BLEU-4, METEOR, ROUGE-L, SPICE, and WMD [43]. They calculate word overlap between candidate and reference sentences and range it between 1-100. Higher values indicated better results.

BLEU (Bilingual Evaluation Understudy) [38] 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 metric 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, the 1 to 4-grams used when computing the metric – the respective variants are called BLEU-1 up to BLEU-4.

METEOR (Metric for Evaluation of Translation with Explicit Ordering) [3] measures the correlation between the predicted caption and human judgment. Compared with BLEU, parts of the sentence are analyzed, not the whole corpus. METEOR algorithm have two stages. At first, tokens from reference captions and candidates are compared. In the second stage final result is calculated. METEOR also analyzes and allows synonyms.

CIDEr (Consensus-based Image Description Evaluation) [50] 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 [41] training, where the strategy is to optimize the model for a specific metric. It results in higher results during the testing phase compared with [41]. Furthermore, CIDEr optimization during training impacts on high scores in BLEU, METEOR and SPICE metrics.

ROGUE-L (Recall-Oriented Understudy for Gisting Evaluation) [30] is a set of metrics: Recall, F1 and precision. Algorithm finds the longest common sequence of tokens between predicted and reference captions. Sequences must be in the same order but not next to each other.

WMD (Word Mover's Distance) [26] is based on a machine learning model to count similarity between texts. Metric is distinguished from others because it measures common sense between texts. It does not investigate the occurrence of tokens. Instead, it measures the semantic length between sentences by counting the probability of the occurrence of synonyms.

All the above metrics are used in various NLP tasks. However, according to some investigations [7], they do not correlate with a human judgment, what makes them not adequate to measure the similarity of image captions [1]. Among the known metrics the one that correlates with the human judgment is SPICE (Semantic Propositional Image Caption Evaluation) [1]. This metric measures similarity between sentences, represented by a directed graph. SPICE algorithm at the beginning creates two directed graphs. First one is for all reference captions and the second is for the candidate sentence. Graphs elements can belong to three groups. 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.

IV. Experimental setup

A. Datasets

There are several datasets used for image captioning. They differ in the number of images and their size, also captions can vary in format and length. Most commonly used are Flickr8k [19], Flickr30k [58] and MSCOCO 2014 [5], [31]. All these sets consist of a number of images with associated captions, usually 5 per image.

Dataset Flickr30k includes 30k images and each photo has five captions. Training set consists of 29k images and 1k is for testing. Flickr8k is a subset of Flickr30k and contains 8k images, with five annotations for each picture. Each caption fully describes a scene and is entirely based on human judgment. In the test split, there are 7k images and the rest of the data is used for testing.

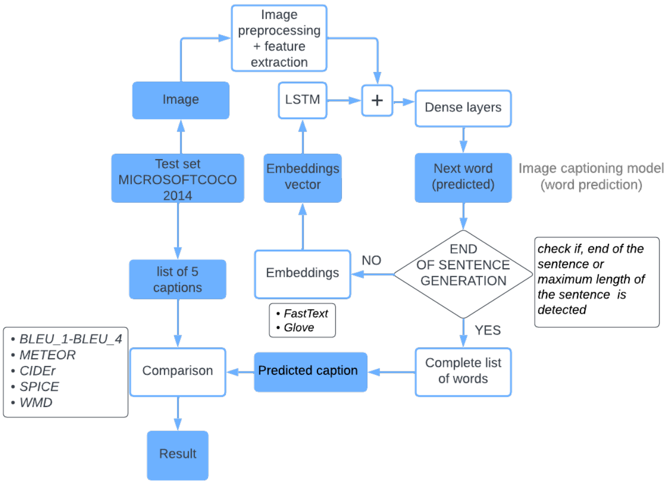


Image preprocessing + feature extraction
Image
LSTM
+
Dense layers
Test set MICROSOFTCOCO 2014
Embeddings vector
Next word (predicted)
Image captioning model (word prediction)
list of 5 captions
Embeddings
NO
END OF SENTENCE GENERATION
check if, end of the sentence or maximum length of the sentence is detected
• FastText
• Glove
YES
• BLEU_1-BLEU_4
• METEOR
• CIDEr
• SPICE
• WMD
Comparison
Predicted caption
Complete list of words
Result

Fig. 2: Diagram of experimental setup.

During our experiments, we used for the evaluation and training the MSCOCO dataset. It consists of more than 120k images from various everyday scenes. Five captions describe each photo in natural language. In the image captioning area, the most popular MSCOCO data partitioning for testing, validation and training purposes is Karpathy split [24], where there are 113k images in training, 5k in validation and 5k in test disjoint subsets.

B. Image preprocessing

Motivated by [24], and considering the variety of available pretrained object detection CNN models and language processing models, we conducted experiments to check how input data to the model can affect the learning process. The whole experimental process involves encoding images and text features simultaneously and generating a final sequence of tokens (caption) word by word during decoding.

Images from the dataset are resized and normalized before entering the image captioning model to be compatible with one of the CNN networks. For VGG16, VGG19, Resnet152V2, Resnet50, DenseNet121, DenseNet201, MobileNet, MobileNetV2 input shape is 224x224x3 and 299x299x3 for InceptionV3, Xception. As a result, we obtained features vectors with the following sizes, corresponding to the preprocessed input image: 4096-element vector for VGG16, VGG19; 2048 for InceptionV3, Xception and Resnet152V2; size 1024 for denseNet121; size 1920 for DenseNet201; 1000 elements for MobileNet; size 1280 for MobileNetV2. We used CNN models pretrained on the ImageNet [42] dataset, where the network's fully connected layers is removed since we do not need

[1] The authors of [7] propose their own metric, but due to much its much lesser (more that 10x) popularity comparing with SPICE, we decided to use that latter in the current study.

the probability distribution on 1000 image categories from ImageNet.

C. Text preprocessing

A separated preprocessing was performed for captions. At the beginning, all words were converted to lowercase, tokenized. We removed punctuations, hanging single-letter words and discarded rare words that occurred less than five times. As a result, we achieved the following vocabularies, also called dictionaries: Flickr8k, Flickr30k, and MSCOCO 2014, that will be used to create embedding matrixes from embedding vectors. Before being handled by LSTM Network, word sequences must be represented in word embeddings vectors. In our model, Glove and FastText have been used as embedding.

Preprocessed captions, consumed by the captioning model, are appended with *start* and *stop* tokens to mark the beginning and end of the sentence, respectively. In the next step, a vocabulary of all words occurring in the captions in the training set is prepared (along with *start* and *stop* tokens). As a result, a dictionary of all words in our corpus is produced to identify tokens by index explicitly. Each generated word is processed by embedding prior to its providing into the LSTM model input.

We adopted pretrained versions of FastText and Glove to extract the text features. We preprocessed sentences from the train and test dataset (described in the previous section) and finally achieved a vocabulary of size 7293. Each word is then embedded to a 200-element vector for Glove and 300-element vector for FastText word embedding space.

D. Training and testing

During training, the model processes combined 256-element vector of word embeddings and image feature vectors based on the CNN model for a given image. At each time step model predicts a word for the processed image and compares it with the ground truth word from the training set, which corresponds to the processed image. Predicted word and ground truth word (from the training set) are compared using the cross-entropy measure (see Fig. 1).

During the testing, image captioning model is fed by a preprocessed photo. 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 served as the embeddings, corresponding to the dictionary. 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 embeddings 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.

E. Evaluation

We investigated the performance of each image encoder, with each text encoder mentioned previously, with BLEU-1 – BLEU-4, METEOR, ROGUE-L, WMD, CIDEr, and SPICE metrics. The complete process is repeated for other CNN architectures and embedding methods to achieve a comprehensive perspective of the performance of different CNN architectures along with different embedding methods. Backbone-embedding pairs tested during experiments are presented in Table. I. The complete process of evaluation is presented in Fig. 2.

For further analysis, we also examined word and bigrams occurences from a training set and predicted captions to determine why some captions are incorrectly generated and what are the collocations of a training set with the parts of the sentence that do not describe the real image content.

V. Results

Table I shows the results of image captioning metrics calculated for different image and text features extractors. We analyzed all models accordingly to the BLEU-1 – BLEU-4, METEOR, ROUGE-L, WMD, CIDEr, and SPICE metrics. Following the literature, to evaluate the performance we used most recent CIDEr and SPICE metrics, keeping the remainder for comparative purposes. For the same purposes we added four reference methods in last four rows of the table.

From the obtained results, we can see that model performance depends mostly on the CNN backbone used. Best results considering the CIDEr metric has been achieved for Xception backbone feature extractor, second place belong 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 from the image backbone feature extractor. The evaluated quality of caption extractors is correlated with the accuracy of backbones. Practically for each metric, the order of models sorted by the metric value is similar to the order of backbones when sorted by accuracy both in top-1 and top-5 variants[2]. 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 embeddings achieve higher results than Glove for the same image features extractor. Which suggests that FastText adapts easier for different CNN models, than Glove. Long feature vectors does not imply higher performance. The longest feature vectors that are generated by VGG backbones does not imply higher values of measures. The winning models are using 2048 (Xception) and 1920 (DenseNet201) vectors. Average time of sequence generation is not correlated with the model complexity (no. of model params). Differences in execution time between models spreads from 874 to 1417 ms. The fastest is DenseNet201, which is also second best model.

Example correct captions obtained by the Xception + Glove pair are given in Table II, the respective images are shown

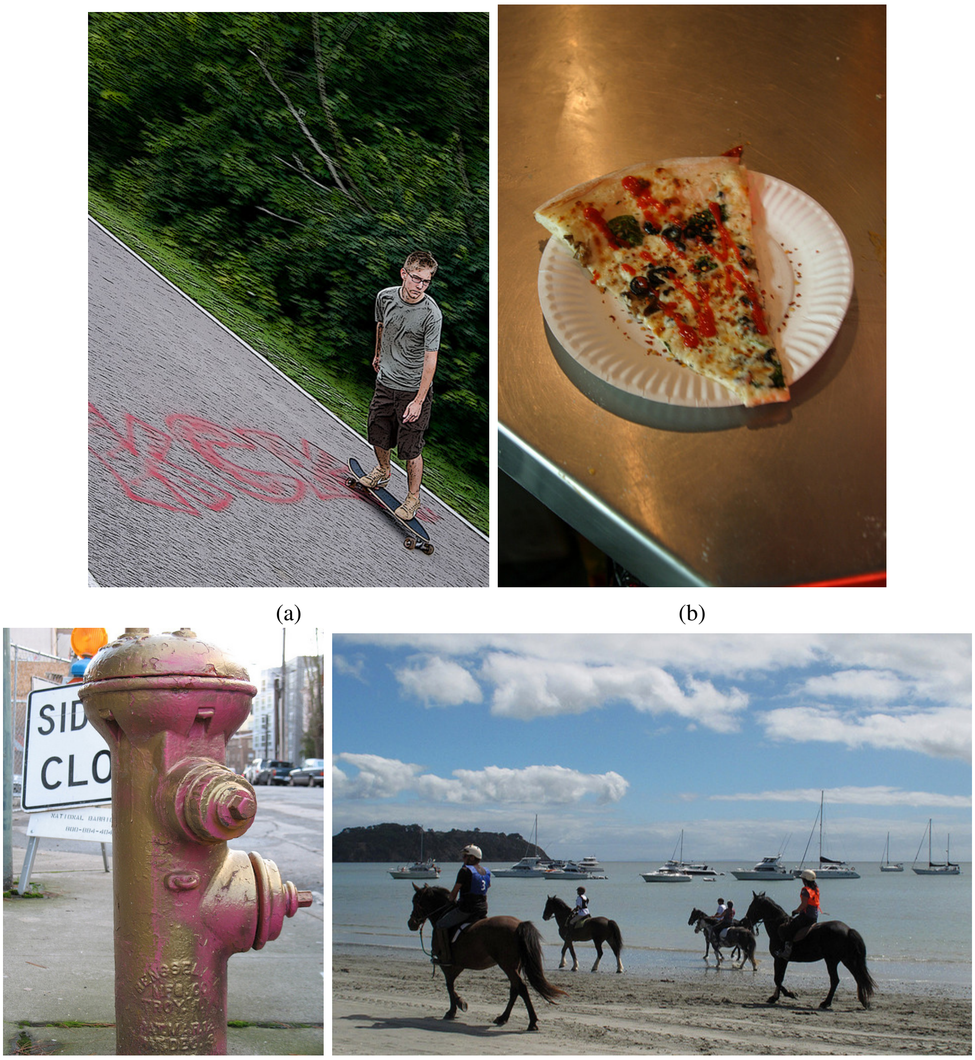
[2] Where top-n means that – in case of complete initial model of the backbone (i.e. model that contains both, the convolutional and fully-connected layers) the proper answer i.e. predicted class is among n-classes of highest output probability.

TABLE I: Evaluation results for MSCOCO 2014 test dataset (5000 images). Metrics' values are averaged over the whole test dataset. Higher results implies better image captioning performance.

Image features	No. of model parameters(mln)	Size of the input image features vector	top-1	top-5	Embeddings	Time of sentence generation (ms)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROGUE-L	WMD	CIDEr	SPICE
Vgg16	14.71	4096	71.3	90.1	FastText	1043	64.47	45.73	31.54	21.86	20.86	47.01	47.43	67.76	13.81
					Glove	1088	64.25	45.62	31.63	22.09	20.78	46.99	47.35	67.35	13.64
Vgg19	20.02	4096	71.3	90.1	FastText	1045	65.42	46.89	32.72	22.93	21.61	48.09	48.40	71.79	14.46
					Glove	1023	64.10	45.83	31.86	22.34	21.11	47.24	47.71	69.62	13.93
Resnet152V2	58.33	2048	78	94.2	FastText	1209	65.28	46.78	32.47	22.61	21.30	47.58	48.04	70.07	14.16
					Glove	1417	64.91	46.78	32.57	22.86	21.38	47.80	48.05	70.77	14.08
Resnet50	23.59	2048	74.9	92.1	FastText	1058	65.97	47.82	33.79	24.02	21.88	48.39	48.79	74.47	14.71
					Glove	1234	65.33	47.26	33.26	23.44	21.65	48.28	48.46	73.12	14.43
InceptionV3	21.8	2048	77.9	93.7	FastText	961	66.15	47.87	33.57	23.63	21.92	48.41	48.92	75.04	14.83
					Glove	980	66.12	47.72	33.35	23.38	21.84	48.20	48.75	74.16	14.72
Xception	20.86	2048	79	94.5	FastText	1026	67.01	48.80	34.45	24.30	22.36	48.85	49.50	77.64	15.18
					Glove	1107	66.59	48.63	34.34	24.33	22.43	48.91	49.43	**78.13**	15.16
DenseNet121	7.04	1024	75	92.3	FastText	1180	65.39	47.09	32.89	23.09	21.60	47.99	48.31	72.36	14.25
					Glove	1234	65.03	47.02	32.96	23.26	21.64	47.87	48.38	71.94	14.13
DenseNet201	18.32	1920	77.3	93.6	FastText	874	66.59	48.73	34.57	24.55	22.25	49.01	49.20	76.74	14.83
					Glove	914	66.35	48.41	34.26	24.18	22.46	49.08	49.29	76.54	14.96
MobileNet	4.25	1000	70.4	89.5	FastText	976	65.02	46.93	32.85	23.02	21.65	47.98	48.15	71.24	14.31
					Glove	965	64.35	46.14	32.12	22.42	21.20	47.45	47.64	69.28	13.76
MobileNetV2	2.26	1280	71.3	90.1	FastText	1072	65.13	47.22	33.17	23.32	21.79	48.22	48.62	73.79	14.62
					Glove	1048	65.39	47.14	33.04	23.24	21.64	47.96	48.35	73.03	14.55
Karpathy [24]							62.50	45.00	32.10	23.00	19.50	-	-	66.00	-
Xu [57]							67.9	49.3	34.7	24.3	22.2	48.8	-	75.4	-
Sugano [46]							71.4	50.5	35.2	24.5	21.9	52.4	-	63.8	-
Lebret [28]							73	50	34	23	-	-	-	-	

TABLE II: Overview of four images with properly predicted captions (Xception image features extractor, Glove embeddings). along with the results of evaluation metrics for them.

Image	Fig. 3a	Fig. 3d	Fig. 3b	Fig. 3c
Ground truth captions	*A young man riding a skateboard down a street. *A man riding a skateboard down a road. *A man skateboards down a steep incline on an area painted with graffiti. *Man on a skateboard crossing over some graffiti *A man riding a skateboard down a hill.	*Horses walk along a beach while boats ride at their moorings offshore. *Some people riding horses on some sand and some boats and water *A group of people riding horses on a beach. *Some people are riding horses along a shoreline. *A group of people riding horses on top of a sandy beach.	*A slice of pizza on a paper plate. *A slice of pizza being served on a plate. *A slice of pizza sits on the paper plate *The metal table has a slice of pizza on a plate. *A slice of pizza is sitting on the top of a paper plate.	*A red and gold painted fire hydrant on the street *A fire hydrant on the side of the road *A multicolored fire hydrant that is on the sidewalk. *A fire hydrant on the side of a street. *A fire hydrant is standing on the sidewalk with two spouts.
Predicted caption	A man riding a skateboard down a street	A group of people riding horses on a beach	A slice of pizza on a plate	A fire hydrant on the side of the street
BLEU-1	100.00	100.00	70.00	100.00
BLEU-2	100.00	100.00	68.31	100.00
BLEU-3	100.00	100.00	66.32	94.99
BLEU-4	100.00	100.00	63.89	91.93
METEOR	41.35	14.28	15.11	28.97
CIDEr	482.15	419.35	411.58	479.30
ROGUE_L	93.13	100.00	79.37	88.89



(a) (b)

(c) (d)

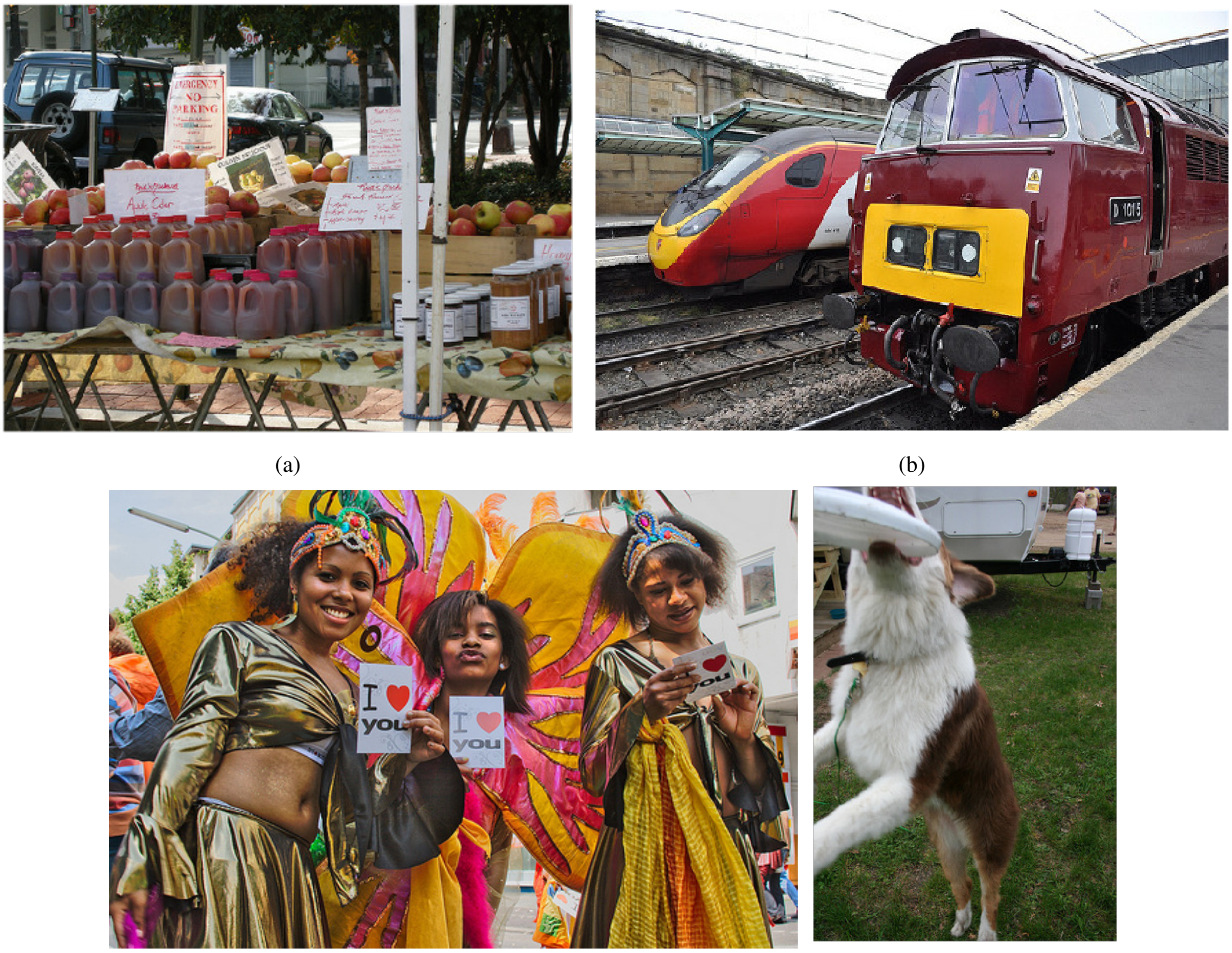
Fig. 3: Images with properly predicted captions (see Table II for details)

in Fig. 3. The table contains ground-truth 5 captions from the dataset metadata, captions obtained from the model and values of metrics. The generated captions sound good, are grammatically correct and consistent with the image content.

In contrast to the above, Table III presents inadequately predicted captions for four images obtained using different methods.

During this experiment, we checked that the resulting captions' wrong parts occur more often in the training set data. For Fig. 4d wrong part of the caption is the *with people standing*. Bigram *with people* occurs 1328 times, *people standing* 2740 times in training set. Those bigrams occur relatively often compared to other parts of the sentence. Also, for Fig. 4b bigrams that form *laying on a couch* occur very often in MSCOCO 2014 training dataset. Especially in the example Fig. 4c, bigrams "front of", "woman holding" are very common in the training dataset.

To explore deeply the possible reasons for incorrect captions, we investigated vocabulary of single words and bigrams used for training. The total size of vocabulary (the number of unique words) equals 26335 for 113350 images described using 5 alternative sentences each, which gives us 566747 captions. The similar numbers for the training set are the following: number of images 5000, of sentences 25000, of unique words: 7197 among which 503 words were used only in the captions in the test set (the remainder i.e. 6694 words are also present in the training set vocabulary). Considering the fact that each of investigated models is being learned on the training set, only words that are present in this vocabulary are used to predict ANY output sentence (correct or not). In case the captions in the test set, the number of words that was not present in the training set equals 503. This implies that, object, actions, situation, scene elements etc. that was described using these words, would never be produced properly (when testing,



(a) (b)

(c) (d)

Fig. 4: Images with improperly predicted captions (see Table III for details)

the words in the test set vocabulary are obviously not used).

For further analysis, we tried to find why parts of the sentence are inadequate and how it is affected by training data. Regarding that, we examined how bigrams from predicted captions compare to those in the training set. We extracted bigrams from the MSCOCO 2014 training set with the number of their occurrences. Then we also extracted bigrams from predicted captions. As a result, we achieved a summary of bigrams in the training dataset and in a set of predicted captions, along with a number of their occurrences. The result for four example images are shown in Fig. III. Not surprisingly, the model, to construct captions, is using more frequent bigrams from the training set.

VI. Conclusions

In this paper, we analyzed how image features and word encoding affect the results of the encoder-decoder image captioning model. Our experiments proved that encoding input data plays in this area the primary role. During our research, we recognized that image captioning involves merging features from different modalities. Because of that, encoding of both image and features must cooperate, so finding the optimal pair for specific model architecture is crucial and we can significantly improve the results of the model predictions with that principle. The influence of the image feature extractor by the CNN backbone is crucial in this type of captioning model, it affects more the performance than the word embedding scheme. The Xception with Glove and DenseNet201 with FastText, according to our experiment are the best combinations of models' components.

The outcome of our research are applicable in all the research works that lead to the developing the optimal encoder-decoder image captioning model.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer, 2016.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

TABLE III: Overview of four images with incorrectly predicted captions. with the results of evaluation metrics for them. Table also consists bigrams of tokens from predicted caption. along with quantity of those bigrams in the training set.

Image	Fig. 4a		Fig. 4b		Fig. 4c		Fig. 4d	
Ground truth captions	*A pile of containers filled with lots of apple juice. *The stand is selling apples and apple cider outside. *Many various sized bottles of apple cider are on the table. *A bunch of apples and cider for sale on a table. *A farm stand selling apple cider and apples.		*Two red and yellow trains parked next to each other. *Two railroad trains with different front cars together *Two yellow and red trains parked on the tracks *Red and yellow trains sitting side by side to each other. *The front of modern commuter trains at the station		*Three women in bright colors and headdresses are holding love message cards *Three women in costume are holding papers that say "I love you" *Girls in bright costumes holding little signs that say I love you. *Three women in elaborate costumes hold up "I Love You" cards. *Three women in colorful costumes holding I love you signs.		*A brown and white dog with a Frisbee in his mouth . *A dog with his front paws off the ground holds a white Frisbee in his mouth in an RV campground . *A white and brown dog jumps up for a white Frisbee . *Dog catching Frisbee . *The brown and white dog is catching a Frisbee in his mouth .	
Model configuration	Resnet152V2 and FastText		Resnet152V2 and FastText		Xception and FastText		InceptionV3 and Glove	
Predicted caption	a market with a bunch of bananas and vegetables		a train is pulling into a station with people standing by it		a woman holding a umbrella in front of a crowd		a dog is laying on a couch with a frisbee	
	n-gram	**quantity**	**n-gram**	**quantity**	**n-gram**	**quantity**	**n-gram**	**quantity**
bigrams	**’and vegetables’**	**977**	’by it’	85	**’front of’**	**12517**	couch with’	449
	’bananas and’	**411**	’into station’	128	’holding umbrella’	63	’dog is’	1305
	’bunch of’	3724	’is pulling’	285	**’in front’**	**12363**	’is laying’	1294
	’market with’	140	**’people standing’**	**2740**	’of crowd’	199	**’laying on’**	**3539**
	’of bananas’	1027	’pulling into’	246	’umbrella in’	328	**’on couch’**	**1410**
	’with bunch’	386	’standing by’	1071	**’Woman holding’**	**1562**	’with frisbee’	853
			’station with’	292				
			’train is’	1310				
			’with people’	**1328**				
BLEU-1	44.44		33.33		19.99		54.29	
BLEU-2	0		28.87		0		46.73	
BLEU-3	0		22.83		0		29.12	
BLEU-4	0		0		0		0	
METEOR	2.24		4.52		10.5		1.65	
CIDEr	0.55		24.75		7.2		86.90	
ROGUE_L	39.29		33.33		0.1		58.65	

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[7] Y. Cui, G. Yang, A. Veit, X. Huang, and S. Belongie. Learning to evaluate image captioning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5804–5812, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

[8] J. Delbrouck and S. Dupont. Bringing back simplicity and lightliness into neural image captioning. *CoRR*, abs/1810.06245, 2018.

[9] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017.

[10] H. Dong, J. Zhang, D. McIlwraith, and Y. Guo. I2t2i: Learning text to image synthesis with textual data augmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, page 2015–2019. IEEE Press, 2017.

[11] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Lawrence Zitnick, and G. Zweig. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[12] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 15–29, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[14] J. Gu, G. Wang, J. Cai, and T. Chen. An empirical study of language cnn for image captioning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1231–1240, 2016.

[15] L. Guo, J. Liu, J. Tang, J. Li, W. Luo, and H. Lu. Aligning linguistic words and visual semantic units for image captioning. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, page 765–773, New York, NY, USA, 2019. Association for Computing Machinery.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[17] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, 04 1998.

[18] S. Hochreiter and J. Schmidhuber. Lstm long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[19] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 05 2013.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] A. Janusz, D. Kałuża, M. Matraszek, Łukasz Grad, M. Świechowski, and D. Ślęzak. Learning multimodal entity representations and their ensembles, with applications in a data-driven advisory framework for video game players. *Information Sciences*, 617:193–210, 2022.

[23] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[24] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

[25] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, number 2 in Proceedings of Machine Learning Research, pages 595–603, Bejing, China, 22–24 Jun 2014. PMLR.

[26] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[27] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 359–368, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[28] R. Lebret, P. O. Pinheiro, and R. Collobert. Phrase-based image captioning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2085–2094. JMLR.org, 2015.

[29] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[30] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[32] S. Liu, L. Bai, Y. Hu, and H. Wang. Image captioning based on deep neural networks. *MATEC Web of Conferences*, 232:01052, 11 2018.

[33] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3242–3250, 2017.

[34] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *CoRR*, abs/1410.1090, 2014.

[35] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[36] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, page 747–756, USA, 2012. Association for Computational Linguistics.

[37] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[39] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[40] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[41] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[43] A. B. Sai, A. K. Mohankumar, and M. M. Khapra. A survey of evaluation metrics used for NLG systems. *CoRR*, abs/2008.12009, 2020.

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[45] R. Subash, R. Jebakumar, Y. Kamdar, and N. Bhatt. Automatic image captioning using convolution neural networks and lstm. *Journal of Physics: Conference Series*, 1362(1):012096, nov 2019.

[46] Y. Sugano and A. Bulling. Seeing with humans: Gaze-assisted neural image captioning. *ArXiv*, abs/1608.05203, 2016.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[49] M. Toshevska, F. Stojanovska, E. Zdravevski, P. Lameski, and S. Gievska. Exploration into deep learning text generation architectures for dense image captioning. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 129–136, 2020.

[50] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015.

[51] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[52] S.-S. Wang and R.-Y. Dong. Learning complex spatial relation model from spatial data. *Journal of Computers*, 30(6):123–136, 2019.

[53] Y. Xian and Y. Tian. Self-guiding multimodal lstm-when we do not have a perfect training dataset for image captioning. *IEEE Transactions on Image Processing*, PP, 09 2017.

[54] X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan. Dense semantic embedding network for image captioning. *Pattern Recognition*, 90:285–296, 2019.

[55] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR.

[56] K. Xu, H. Wang, and P. Tang. Image captioning with deep lstm based on sequential residual. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 361–366, 2017.

[57] N. Xu, A. Liu, J. Liu, W. Nie, and Y. Su. Scene graph captioner: Image captioning based on structural visual representation. *J. Vis. Commun. Image Represent.*, 58:477–485, 2019.

[58] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[59] X. Zhang, S. He, X. Song, R. W. Lau, J. Jiao, and Q. Ye. Image captioning via semantic element embedding. *Neurocomputing*, 395:212–221, 2020.