Enhanced Image Captioning using Efficient Pretrained Word Embedding

Quang-Huy Nguyen , Hoai-Phong Le , Xuan-Nam Cao , Minh-Triet Tran
Falcuty of Information Technology, University of Science, VNU-HCM
Vietnam National University, Ho Chi Minh City, Vietnam
20120497@student.hcmus.edu.vn, 20120545@student.hcmus.edu.vn

***Abstract*—Image captioning is an interesting and challenging problem at the intersection of computer vision and natural language processing. One of the most significant breakthroughs of natural language processing is the use of vector representations for words, representing them as numerical vectors rather than as sequences of characters. These representations can be trained as task-specific or pretrained using a large corpus of text. However, in the context of image captioning, the use of pretrained word embeddings has not been widely explored. In this paper, we investigate the impact of pretrained word embeddings on the quality of an image captioning model. Specifically, we focus on ResNet-50 and LSTM with a soft attention mechanism approach and experiments with three different word embeddings: non-pretrained, pretrained fastText, and pretrained GLoVe. The evaluation is conducted on the Flickr30k dataset, which comprises 31,000 images, each associated with five captions. We report that the BLEU-4 scores obtained are 0.2407 for the non-pretrained embedding, 0.2412 for fastText, and 0.2475 for GLoVe. The results indicate that both fastText and GLoVe embeddings enhance the model's ability to generate more accurate and contextually relevant captions. Overall, this work demonstrates that using pretrained word embeddings slightly improves the performance of the captioning model, with GLoVe embeddings being particularly well suited for this image captioning task.**

***Index Terms*—Image captioning, word embedding, fastText, GLoVe**

I. Introduction

Image captioning is a challenging and meaningful domain of study that has attracted extensive research. It takes an image as an input and generates a textual description of the image as the output. Image captioning has many real-world applications, including human-computer interaction [42], [78], medical image captioning [24], [29], quality control in industry [43], image search engines [26], and assistive technologies for visually impaired individuals [12], [33], [77]. Owing to its practical significance, there is a pressing need for continuous improvements in image captioning models.

A quick glance at an image enables a human to effortlessly identify and describe an immense amount of details in the visual scene [16]. However, image captioning has long been regarded as a difficult problem in computer science because it necessitates the integration of two major fields of Artificial Intelligence [57]: Computer Vision [25], [68] and Natural Language Processing [8], [9].

The task requires the model to not only accurately identify objects within images but also to comprehend the interactions and relationships between these objects to generate meaningful captions. Determining the presence, attributes, and relationships of objects in an image is not an easy task itself. Generating a human-like sentence to describe such information makes this task even more difficult [4].

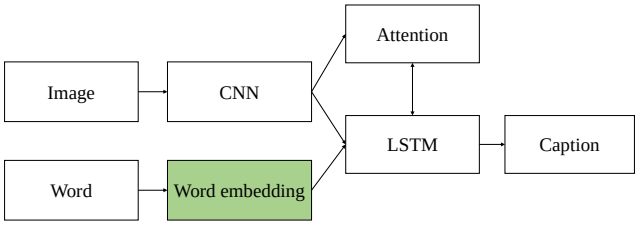


Fig. 1: Block diagram of an image captioning model using an encoder-decoder framework with an attention mechanism. This paper uses this architecture and focuses on Word embedding block.

Fig. 2: Examples of captions for given images. These captions are generated using the proposed model with GloVe pretrained word embedding.

In the early stages of image captioning research, traditional image captioning approaches often rely on feature extraction methods, such as Histogram of Oriented Gradients (HOG) [62], Scale Invariant Feature Transform (SIFT) [10], Local Binary Patterns (LBP) [50], and Bag of Words (BoW) [63]. These methods have demonstrated impressive results and have become popular in the field for many years. However, with the complexities involved in feature extraction, traditional methods are now less favored than deep learning-based approaches, which have the ability to automatically learn features from large datasets [55].

The rapid advancement of deep learning [6], [36], [40] has revolutionized image captioning using data-driven neural network approaches. Encoder-decoder models have been widely used in image captioning [70]. Convolutional Neural Networks (CNNs) [1], [32], [39] are employed as encoders to extract visual features from images, whereas Recurrent Neural Networks (RNNs) [53] or variants such as Long Short-Term Memory (LSTM) [31] and Gated Recurrent Units (GRU) [11] serve as decoders for language sequences. This powerful combination helps the model generate descriptive and contextually relevant captions for diverse visual content.

At each step of the decoder's word generation process, it is unnecessary to consider information from the entire image; only information from the relevant regions is sufficient. This idea was proposed by Xu et al. [71], and it led to the introduction of the Attention mechanism [49]. It enables the model to focus on the necessary regions of the image during each word-generation step, thereby enhancing the accuracy of the model. Additionally, the Attention mechanism improves the interpretability of the model [17], allowing us to visualize the regions of the image that the model focuses on during the caption generation process.

To generate word sequences, whether using RNNs or transformers [64], all methods operate not on sequences of characters, but at a higher level of abstraction, representing words as vectors of numbers. These vectors can be pre-associated with each word or adjusted during model training. Several techniques exist for establishing correspondence between words and numeric vectors. While one-hot encoding [58] is a straightforward approach, more sophisticated methods, such as Word2vec [46] or GLoVe [54], have been developed.

In the context of image captioning, pretrained vector representations for words have not been widely utilized because models are usually trained from scratch during the training process to generate textual descriptions. In this work, we explore the potential benefits of incorporating pretrained word embeddings, such as fastText and GLoVe, to enhance the performance of the model.

To gain insight into the impact of pretrained word embeddings on the quality and performance of the image captioning models, we constructed three models: one without using pretrained embeddings, another utilizing fastText pretrained embeddings, and the third employing GLoVe pretrained embeddings. All of these models adopt an encoder-decoder architecture with a soft attention mechanism. The evaluation is performed on the Flickr30k dataset [28].

The experiment yields significant results, demonstrating the advantages of utilizing pretrained GLoVe embeddings in comparison to models without pretrained embeddings. We observe a remarkable 2.82% improvement in BLEU-4 scores and a 2.41% enhancement in ROUGE-L scores. These findings underscore the potential and effectiveness of using pretrained word embeddings to address the image captioning problem.

The key contributions of this study are as follows:

(1) We introduce a novel approach using pretrained word embedding for image captioning, which enhances the model's performance.

(2) We conduct thorough experiments on the Flickr30k dataset and evaluate by using multiple evaluation metrics, including BLEU-n, ROUGE, SPICE, and METEOR, to comprehensively measure the performance and compare the impact of pretrained embeddings on the model's caption quality.

(3) We perform a comparative analysis to highlight the significance of the pretrained word embeddings. We investigate how each type of pretrained embedding, including fastText and GLoVe, influences the output of the image captioning model. This analysis provides valuable insights into the advantages and limitations of utilizing pretrained word embeddings in the image captioning context.

The remainder of this paper is organized as follows. Sec.II reviews related work on image captioning approaches and word embedding. Sec.III.1/ describes our proposed solutions, including encoder-decoder architecture (Sec.III.2/ and Sec.III.3/) with soft attention mechanism (Sec.III.4/) and GLoVE and fastText word embeddings (Sec.III.5/). Sec.IV presents the experimental results that demonstrate the effectiveness of our method. Finally, Sec.V concludes our study and paves the way for future work.

II. Related work

In this section, we provide relevant background on previous studies on image caption generation. We also categorize various approaches and introduce the enhancements achieved through word embedding.

A. Image captioning

With the recent surge of research interest in image captioning, a large number of approaches have been proposed. Based on the techniques adopted in each method, these research endeavors can be broadly classified into different categories, as presented in Table I. We classify these approaches into three main categories: template-based, retrieval-based, and neural network-based approaches. .

The retrieval-based method is commonly used in the early work of image captioning. This approach uses prespecified sentence pools to produce captions for query images, either by retrieving existing sentences or composing new ones from the retrieved set. Ordonez et al. [51] first employed global image descriptors to retrieve a set of images from a web-scale collection of captioned photographs. Hodosh et al. [23] employed the Kernel Canonical Correlation Analysis technique [3], [20] to project image and text items into a common space, where the training images and their corresponding captions are maximally correlated. These methods generated general and syntactically correct captions. However, it is difficult to generate image-specific and semantically correct captions.

The template-based method first extracts the local features of the image, captures the object's category and position in the image, and then fills the predesigned sentence template with the words corresponding to these pieces of information. For example, Farhadi et al. [15] used a triplet of scene elements to fill template slots for generating image captions. Mitchell

TABLE I: Summary of image captioning methods

Approach		**Representation methods**
Retrieval based		Ordonez et al. [51], Hodosh et al. [23], Gupta et al. [19], Kuznetsova et al. [38]
Template based		Farhadi et al. [15], Kulkarni et al. [37], Yang et al. [73], Li et al. [41], Mitchell et al. [47]
Neural networks based	Augmenting early work by deep model	Socher et al. [59], Karpathy et al. [30], Ma et al. [44], Yan and Mikolajczyk [72]
	Encoder–decoder framework	Vinyals et al. [65], Kiros et al. [35], Donahue et al. [13], Jia et al. [28], Wu et al. [69]
	Attention mechanism guided	Xu et al. [71], You et al. [76], Yang et al. [74]

et al. employed algorithms to process and represent an image using $\langle objects, actions, spatial\, relationships \rangle$ triplets [47]. Kulkarni et al. adopted a Conditional Random Field method to determine the image content before filling in the gaps [37]. Template-based methods can generate grammatically correct captions. However, templates are predefined, and the length of the captions cannot be varied.

With significant advancements in deep learning [6], recent studies begins to rely on deep neural networks for automatic image captioning, replacing hand-engineered features and shallow models. For instance, Socher et al. [59] used dependency-tree recursive neural networks for phrase and sentence representation, and a visual model based on a deep neural network for image feature extraction. Karpathy et al. [30] propose embedding sentence and image fragments into a common space for sentence ranking. Although deep neural networks enhance performance, the limitations of retrieval- and template-based methods persist.

Inspired by machine translation [60], some researchers have considered image captioning as a translation problem [13]. Kiros et al. [35] were among the pioneers in introducing the encoder-decoder framework for image captioning, utilizing a deep CNN to encode visual information and LSTM to encode text data. Vinyals et al. [65] proposed a model based on GoogleNet [61] and LSTM, simplifying the process by feeding the entire image feature into the initial time step of the LSTM. To generate captions closely related to image content, Jia et al. [28] introduced g-LSTM, which extracts semantic information from the image to represent the relationship between the image and the caption, guiding each time step of LSTM generation.

A major limitation in the encoder-decoder approach of image captioning is the failure to fully utilize all image details while using convolutional networks to extract features. To address this, Xu et al. [71] proposed an attention-based approach. They divide the image into 14×14 image blocks evenly and use "soft" and "hard" attention to automatically focus on salient regions of the image for caption generation. Attention-based approaches have demonstrated remarkable performance improvements compared to conventional methods. By incorporating attention mechanism, the model can prioritize important visual cues, objects, and contexts while generating captions, resulting in more coherent and contextually relevant captions.

In this study, we explore an encoder-decoder architecture with a soft-attention mechanism similar to [71].

B. Word Embedding

Word embedding is another enhancement for improving the language decoder in image-captioning models [14]. It acts as a distinct layer that transforms input word sequences into numerical representations and effectively maps them to their corresponding positions in the vocabulary. The weights of this embedding layer are learned through back-propagation or adapted from pretrained language models.

There are a few researchers on this point, such as Quanzeng et al. [76], who utilized pretrained GLoVe word vectors from Pennington et al. [54] to encode word information into LSTM. Embedding GLoVe enhances the skip-gram model of Mikolov et al. [46], but both rely on the linear sequential properties of the text. To put it another way, both approaches use the text's word co-occurrence properties to generate text representations.

In contrast, Vinyals et al. [66] found that using pretrained embeddings did not lead to improved model performance. However, Atliha et al. [2] demonstrated that utilizing pretrained GLoVe embeddings with fine-tuning can significantly enhance model performance and is especially beneficial for improving the training of image captioning models.

In this study, we construct three models with the same architecture but different word embeddings: no pretrained embedding, pretrained fastText, and GLoVe. By comparing these models, we demonstrate that using pretrained word embeddings leads to higher effectiveness and performance.

III. Proposed Method

A. Model overview

To compare the models, we have implemented the Show, Attend and Tell architecture [71], which is an encoder-decoder framework with an Attention mechanism. Because it is a simple classical architecture, we can study the impact of using pretrained word embeddings on the performance of the image captioning model.

The architecture consists of three main components, depicted in figure 3

a) Encoder: A convolutional neural network (CNN) responsible for extracting features from the input image. The CNN outputs a fixed-size feature vector that represents the image.

b) Attention Mechanism: This component assigns weights to image features to indicate their importance. The attention mechanism allows the model to focus on the relevant

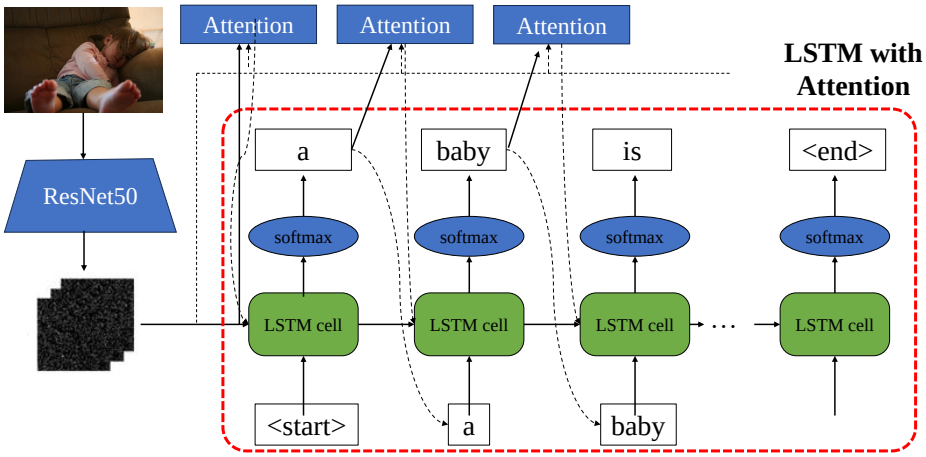


Fig. 3: ResNet50 - LSTM - Attention architecture for image captioning

regions of the image during the caption-generation process. Specifically, the model employs a soft attention mechanism to establish a correlation between the image features and each word in the caption.

c) Decoder: A recurrent neural network (RNN), specifically a Long Short-Term Memory (LSTM) network, uses the output of the CNN as input. LSTM generates each word in the caption based on the previous input and information from the image.

B. Encoder: Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are widely employed as encoders for image captioning tasks. The CNN architecture allows hierarchical feature learning through convolutional operations, which capture local patterns and gradually aggregate them to form higher-level representations. The CNN model is a feedforward neural network in which layers are interconnected through convolution operations.

A common issue in CNN is the vanishing gradient problem, which can hinder the training process and limit the network's ability to learn effectively. The depth of the network is crucial for neural networks, but deeper networks are more difficult to train. To address this challenge, we decide to use the ResNet50 [21].

ResNet [67] introduces the concept of residual blocks, which enable the network to learn identity mapping. The residual block architecture is illustrated in Fig. 4. These residual blocks allow the model to skip connections and propagate information more effectively through the network, thereby mitigating the vanishing-gradient problem. This approach not only boosts the model's performance but also helps accelerate convergence during training.

This model produces a set of feature vectors, specifically L vectors, where each vector belongs to the R^D space, representing different parts within an image. To extract these feature vectors from the layers, the CNN model eliminates Fully Connected layers, which are typically used for object classification tasks.

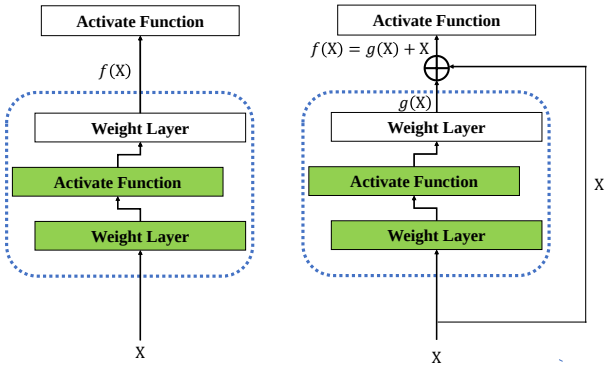


Fig. 4: Different between regular block (left) and residual block (right) (Source: adapted from [27])

C. Decoder: Long Short-term Memory Network

The decoder generates captions for each word based on the features extracted by the encoder. Recurrent Neural Networks (RNNs) are suitable for this task. However, traditional RNNs encounter challenges, such as vanishing and exploding gradient problems. To overcome these issues, Long Short-Term Memory Networks (LSTMs) [22] have been employed.

LSTM offers an effective solution by introducing a specialized structure known as the LSTM cell, which incorporates forget gates to control information flow. These forget gates enable the LSTM to retain relevant information from the previous time steps while discarding irrelevant information. This mechanism ensures that the gradients are more stable during backpropagation, allowing the LSTM to effectively capture long-range dependencies and improve the quality of the generated captions.

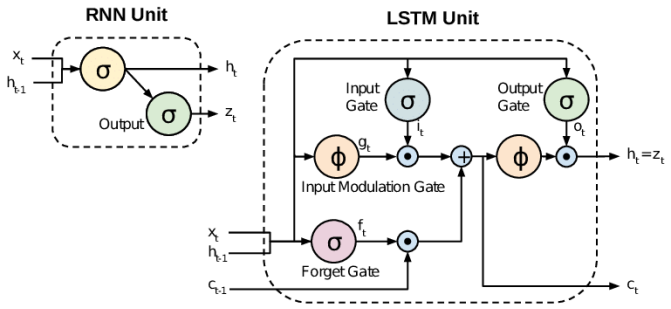


Fig. 5: Diagram of a basic RNN cell (left) and an LSTM memory cell (right) (Source: [56])

An LSTM unit has a more complex structure than an RNN unit. It utilizes gated mechanisms, known as forget gates, to control information flow. Thus, LSTM can selectively retain information. LSTM can effectively store and recall information over long periods, enabling it to learn long-range dependencies.

The formula for updating the cell state at time t in an LSTM unit is as follows:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} \circ W \begin{pmatrix} h_{t-1} \\ x \end{pmatrix}$$

$$c_t = f \circ c_{t-1} + i \circ g$$

$$h_t = o \circ tanh(c_t)$$

$$p_t = softmax(h_t)$$

where, σ is the sigmoid function. i, f, o, c, h are the input, forget, output, memory, hidden states. p_t is the probability distribution for all the words. The word with the highest probability is chosen at each step and included in the next step to form a sentence.

D. Attention Mechanism

The current CNN-LSTM model is limited in that it only provides image content once, specifically at the initial step of the LSTM, attempting to decode the entire image from the last hidden state h_0. While one approach to addressing this limitation involves feeding the entire image information at each step, this method leads to computational inefficiency.

A more effective solution is to focus selectively on important regions within an image by employing a soft attention mechanism [45]. Soft attention has been extensively used in image classification tasks [67] because it allows the model to assign weights based on the significance of different regions. The soft attention mechanism is incorporated by adding an attention gate to the LSTM, enabling the model to concentrate its attention selectively, as shown in Figure 3.

The new formula for updating the cell state at time t is as follow:

$$\begin{pmatrix} i \\ f \\ o \\ g \\ \alpha_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \\ softmax \end{pmatrix} \circ W \begin{pmatrix} I \circ \alpha_{t-1} \\ h_{t-1} \\ x \end{pmatrix}$$

where, α_t is an attention vector at time t.

By using the softmax function, we ensure that each $\alpha_t > 0$ for every pixel and for a vector containing all pixels, $\sum \alpha_t = 1$. Soft attention is differentiable and can be trained using a standard backpropagation algorithm.

To calculate the specific value of α_t for each position i, [71] presents the following equations:

$$e_{ti} = f_{att}(a_i, h_{t-1})$$

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

where, f_{att} is the attention model that uses a multilayer feedforward neural network, α_i represents the degree of focus placed on the position of the model. Subsequently, the context vector $\hat{z}_t$ is computed as follows:

$$\hat{z}_t = \sum_{i=1}^{L} \alpha_{ti} a_i$$

The loss function used in the training process is:

$$L = -\log(P(y|x)) + \lambda \sum_{i}^{L} (1 - \sum_{t}^{C} \alpha_{ti})^2$$

E. Word embedding

1) GLoVe Embeddings: GLoVe (Global Vectors for word representation) [54] is an unsupervised method for obtaining vector representations of words based on the statistics of the co-occurrence of word pairs in a corpus. It combines the benefits of two main model families: global matrix factorization and local context window methods. Therefore, it can produce high-quality word vectors for use in downstream tasks.

2) fastText: fastText[1] provides two efficient word representation learning models based on the skipgram model and the continuous bag-of-words (CBOW) model [46]. These models have been optimized to generate superior word representations by taking advantage of subword information and position weights. The skip-gram model with subword information is introduced in [7], each word is decomposed into its character ngrams and word vectors are the sum of its character ngrams vector representations. The usage of position weights in the CBOW model is introduced in [48], each word vector is element-wise multiplied by a position-dependent vector.

[1]Available at https://github.com/facebookresearch/fastText

IV. Experiments

A. Training Procedure

Models were trained in Google Colaboratory, which provides a NVIDIA Tesla T4 GPU, an Intel Xeon CPU with 2 vCPUs (virtual CPUs), and 13GB of RAM for chargeless access. We used Pytorch to implement the code for these models.

For preprocessing captions, we added four special tokens $< start >$, $< end >$ standing for the start of the sentence, the end of the sentence, $< unk >$ standing for words which appear less than five times, $< pad >$ standing for a pad in the sentence to make all sentences have the same length. Next, we built a vocabulary from captions in the dataset. For pretrained word embeddings, we just kept words that appear in the vocabulary. Images are preprocessed to use Pytorch pretrained models. The detail of the image preprocessing method is described in its weight documentation[2].

We used pretrained ResNet-50 model for encoder which was trained for 1.2 million images of the ImageNet dataset for classification task. Since the pretrained model was trained for classification task, we stripped last two layers to get 14x14x2048 feature map for each image. Subsequently, the LSTM model operates on the flattened 196x2048 encoder output to generate sentences. Each LSTM cell has 512 hidden elements.

For training the model without pretrained word embedding, elements of word embedding matrix are randomly generated from a uniform distribution for easier convergence. For training two models with pretrained word embedding, we used value from pretrained word vectors for initial elements of word embedding matrix. Beam search size 5 was used to produce better captions.

Adam algorithm [34] was used to optimize training process with initial learning rate 10^{-4} for encoder and $4 \cdot 10^{-4}$ for decoder. In addition, we applied dropout technique with a value of 0.5 in LSTM layers to help avoid overfitting.

We trained the model in two stages. At the first stage, we trained only decoder with a batch size of 64 around 15 epochs. Next, we continued from the model has highest BLEU-4 score from the first stage allowing fine-tuning of the encoder with a batch size of 32 around five more epochs. We saved the model having highest BLEU-4 score during validation.

B. Dataset

We evaluate the performance of the proposed model using the Flickr30k dataset [28]. This dataset contains 31,000 images. Each image corresponds to five artificially generated descriptions. The images in this dataset are mainly related to humans involved in everyday activities and events.

We use the Karpathy split [30] to create the training, validation, and test sets. The numbers of images in the training, validation, and test set are 29000, 1000, and 1000, respectively. Each image has five captions provided by human annotators [28].

[2]https://pytorch.org/vision/main/models.html

C. Evaluation Metrics

Unlike typical tasks, in the image captioning problem, in addition to evaluating the accuracy of the generated captions, it is essential to assess their coherence and grammatical correctness.

To evaluate the performance of the models, we use the following metrics: BLEU [52], METEOR [5], and ROUGE-L.

BLEU: This widely used machine translation metric analyzes the co-occurrences of n-grams between candidate (generated) and reference (ground truth) sentences.

METEOR: Based on the harmonic average of the precision and recall, METEOR addresses certain issues present in BLEU. This highly correlates with human discrimination based on recall.

ROUGE-L: This metric relies on the longest common subsequence (LCS) between generated and reference sentences. Higher quality summaries are determined by longer LCS. One drawback of this method is that it requires continuous n-grams.

D. Pretrained word embeddings

1) GLoVe: We used pretrained 300-dimensional word vectors [3] which are trained on the combination of 2014 Wikipedia and Gigaword 5 with total of 6 billion tokens and a vocabulary of the top 400,000 most frequently occurring words using the GLoVe model [54].

2) fastText: We used pretrained 300-dimensional English word vectors provided by [18], which are trained on Common Crawl and Wikipedia using fastText. The model was trained CBOW with position-weights, with character n-grams of length 5, a window of size 5 and 10 negatives.

E. Results and Discussion

1) Quantitative Results:

First, we initiated a comparison between the model using word embeddings trained from scratch and the models utilizing pretrained word embeddings. A graph illustrating the metric values for each training epoch on the validation dataset is shown in Fig. 6.

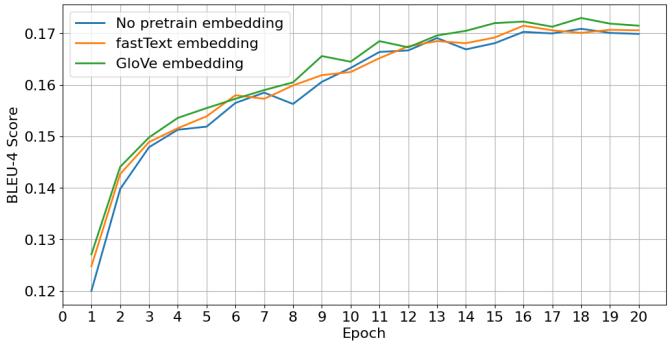
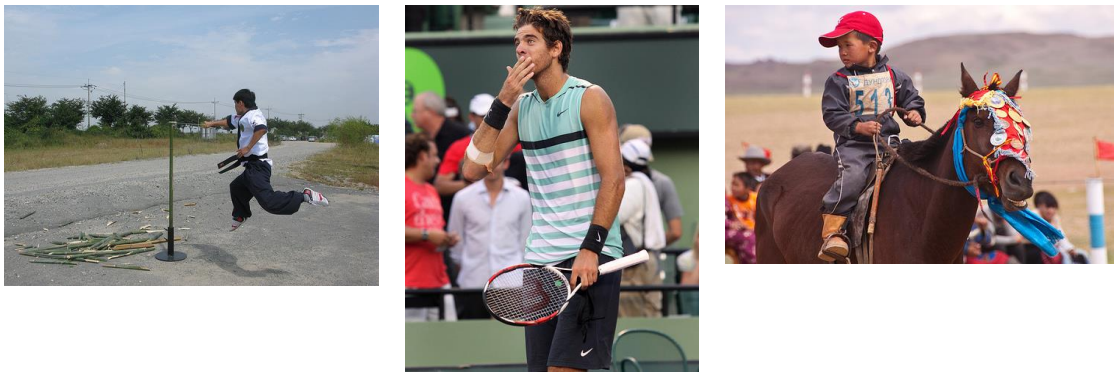


Fig. 6: Validation of BLEU-4 score per epoch during training phase (20 first epochs) for model without pretrained embedding, model with fastText pretrained embedding, and model with GLoVe pretrained embedding.

[3]Available at https://nlp.stanford.edu/projects/GLoVe/

TABLE II: Performance of three models comparison in terms of BLEU-n (n = 1,2,3,4), METEOR, and ROUGE-L on the Flickr30k dataset. Bold numbers represent the best result for each metric.

	BLEU					
Model	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**
No pretrained	65.55	47.39	33.86	24.07	19.51	45.18
fastText	66.19	47.73	33.84	24.12	19.50	45.48
GLoVe	**66.39**	**48.13**	**34.65**	**24.75**	**19.81**	**46.27**



(No pretrained) a man in a white shirt is skateboarding.
(fastText) a man in a white t shirt and blue jeans is jumping in the air
(GLoVE) a man in a white shirt and black pants is jumping in the air

(No pretrained) a woman is playing tennis
(fastText) a man in a striped shirt is playing tennis
(GLoVE) a man in a striped shirt is playing tennis

(No pretrained) a man in a cowboy hat is riding a horse
(fastText) a man in a red hat is riding a horse
(GLoVE) a little boy in a red hat is riding a horse

Fig. 7: Examples of generated captions for given images. Each image has three captions, corresponding to a model that does not use pretrained embedding, a model using fastText embedding, and a model using GloVE embedding.

The results show that models using pretrained embeddings demonstrate significantly higher scores from the initial epoch. In contrast, the model without pretrained embeddings achieves a score of only 0.12 after the first training epoch, whereas the models with fastText and GLoVe pretrained embeddings achieve scores of 0.1248 and 0.1271, respectively. This disparity can be attributed to the fact that pretrained embeddings inherently carry substantial information about word semantics in the language, enabling the model to capitalize on this advantage from the outset.

As time progresses, the performance gap between models with pretrained embeddings and those trained from scratch diminishes, although models with pretrained embeddings still attain superior results. For example, after the 20th epoch, the model without pretrained embeddings achieves a score of 0.1699, whereas the models with pretrained embeddings achieve scores of 0.1706 and 0.1715, respectively, for fastText and GLoVe embeddings.

Additionally, it is important to highlight that the model using GLoVe embeddings outperforms the model utilizing fastText embeddings. The highest-scoring model with fastText pretrained embeddings achieves a score of 0.1707, whereas the best model with GLoVe pretrained embeddings achieves a superior score of 0.1730.

The final results of the comparison between the three models for all metrics using the test dataset are summarized in Table II. The findings clearly demonstrate that models employing pretrained word embeddings consistently yield better performance than models trained from scratch. In particular, the model utilizing the GLoVe embeddings achieves the most impressive results. The use of pretrained GLoVe embeddings leads to notable improvements when compare with the performance of models without pretrained embeddings, with a 2.82% boost in BLEU-4, 1.54% in METEOR, and 2.41% in ROUGE-L.

2) Qualitative Results:

Figure 7 shows three examples of the generated captions for the provided images. Clearly, models utilizing pretrained word embeddings deliver more accurate and natural sounding responses, resembling human-like descriptions. The results emphasize the superiority of GLoVe word embeddings over fastText in improving image captioning performance.

V. Conclusion

In this paper, we discussed the impact of using pretrained word embeddings on the results of the image captioning problem using BLEU, ROUGE-L, and METEOR metrics. The results of the experiment showed that pretrained word embeddings can improve the performance of the model. We found that using pretrained word embeddings with GLoVe yielded significantly better results. We hope that the results of this study will encourage further research on using word embeddings for other tasks in natural language processing. Furthermore, pre-trained word embeddings can be tested in other architectures, such as the object detection + CNN + LSTM approach. [75].

In future work, we will explore fine-tuning strategies for pretrained word embeddings, investigate multimodal embeddings, integrate attention mechanisms, and explore transfer-learning techniques. In addition, we will conduct fine-grained evaluations and test our model on larger, diverse datasets to assess its generalization capability. These efforts aim to enhance image captioning with efficient pretrained word embeddings and advance the field.

References

[1] Viktar Atliha and Dmitrij Sesok. Comparison of VGG and ResNet used as encoders for image captioning. In *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE, April 2020.

[2] Viktar Atliha and Dmitrij Šešok. Pretrained word embeddings for image captioning. In *2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pages 1–4. IEEE, 2021.

[3] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of machine learning research*, 3(Jul):1–48, 2002.

[4] Shuang Bai and Shan An. A survey on automatic image caption generation. *Neurocomputing*, 311:291–304, 2018.

[5] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[10] Lowe David. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[11] Tiago do Carmo Nogueira, Cássio Dener Noronha Vinhal, Gélson da Cruz Júnior, and Matheus Rudolfo Diedrich Ullmann. Reference-based model using multimodal gated recurrent units for image captioning. *Multimedia Tools and Applications*, 79:30615–30635, 2020.

[12] Pierre Dognin, Igor Melnyk, Youssef Mroueh, Inkit Padhi, Mattia Rigotti, Jarret Ross, Yair Schiff, Richard A Young, and Brian Belgodere. Image captioning as an assistive technology: Lessons learned from vizwiz 2020 challenge. *arXiv preprint arXiv:2012.11696*, 2020.

[13] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[14] Samar Elbedwehy, T Medhat, Taher Hamza, and Mohammed F Alrahmawy. Enhanced image captioning using features concatenation and efficient pre-trained word embedding. *Computer Systems Science & Engineering*, 46(3), 2023.

[15] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 15–29. Springer, 2010.

[16] Li Fei-Fei, Asha Iyer, Christof Koch, and Pietro Perona. What do we perceive in a glance of a real-world scene? *Journal of vision*, 7(1):10–10, 2007.

[17] Yuan Gao and Yingjun Ruan. Interpretable deep learning model for building energy consumption prediction based on attention mechanism. *Energy and Buildings*, 252:111379, 2021.

[18] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[19] Ankush Gupta, Yashaswi Verma, and C Jawahar. Choosing linguistics over vision to describe images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 26, pages 606–612, 2012.

[20] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[24] Jia-Hong Huang, Ting-Wei Wu, and Marcel Worring. Contextualized keyword representations for multi-modal retinal image captioning. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 645–652, 2021.

[25] Earnest Paul Ijjina and Chalavadi Krishna Mohan. Hybrid deep neural network model for human action recognition. *Applied soft computing*, 46:936–952, 2016.

[26] Sethurathienam Iyer, Shubham Chaturvedi, and Tirtharaj Dash. Image captioning-based image search engine: An alternative to retrieval by metadata. In *Soft Computing for Problem Solving: SocProS 2017, Volume 2*, pages 181–191. Springer, 2019.

[27] Abbas Jafar and Myungho Lee. High-speed hyperparameter optimization for deep resnet models in image recognition. *Cluster Computing*, pages 1–9, 2021.

[28] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision*, pages 2407–2415, 2015.

[29] Paul Karoly and Linda S Ruehlman. Psychological "resilience" and its correlates in chronic pain: findings from a national community sample. *Pain*, 123(1-2):90–97, 2006.

[30] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[31] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[32] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[33] Young-Bum Kim, Yunyao Li, and Owen Rambow. Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies: Industry papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, 2021.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[37] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2891–2903, 2013.

[38] Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, 2:351–362, 2014.

[39] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[41] Siming Li, Girish Kulkarni, Tamara Berg, Alexander Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228, 2011.

[42] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

[43] Ren C Luo, Yu-Ting Hsu, Yu-Cheng Wen, and Huan-Jun Ye. Visual image caption generation for service robotics and industrial applications. In *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, pages 827–832. IEEE, 2019.

[44] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE international conference on computer vision*, pages 2623–2631, 2015.

[45] Levi McClenny and Ulisses Braga-Neto. Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*, 2020.

[46] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[47] Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alexander Berg, Tamara Berg, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756, 2012.

[48] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in neural information processing systems*, 26, 2013.

[49] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[50] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *Computer Vision-ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part I 6*, pages 404–420. Springer, 2000.

[51] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

[52] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[53] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[54] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[55] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[56] Aliaa Rassem, Mohammed El-Beltagy, and Mohamed Saleh. Cross-country skiing gears classification using deep learning. *CoRR*, abs/1706.08924, 2017.

[57] Antonio M Rinaldi, Cristiano Russo, and Cristian Tommasino. Automatic image captioning combining natural language processing and deep neural networks. *Results in Engineering*, 18:101107, 2023.

[58] Pau Rodríguez, Miguel A Bautista, Jordi Gonzalez, and Sergio Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75:21–31, 2018.

[59] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[60] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[61] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[62] Carlo Tomasi. Histograms of oriented gradients. *Computer Vision Sampler*, pages 1–6, 2012.

[63] Chih-Fong Tsai. Bag-of-words representation in image annotation: A review. *International Scholarly Research Notices*, 2012, 2012.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[65] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[66] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016.

[67] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[68] Shitong Wang, Yizhang Jiang, Fu-Lai Chung, and Pengjiang Qian. Feedforward kernel neural networks, generalized least learning machine, and its deep learning with application to image classification. *Applied Soft Computing*, 37:125–141, 2015.

[69] Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton Van Den Hengel. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1367–1381, 2017.

[70] Xinyu Xiao, Lingfeng Wang, Kun Ding, Shiming Xiang, and Chunhong Pan. Deep hierarchical encoder–decoder network for image captioning. *IEEE Transactions on Multimedia*, 21(11):2942–2956, 2019.

[71] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[72] Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3441–3450, 2015.

[73] Yezhou Yang, Ching Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 444–454, 2011.

[74] Zhilin Yang, Ye Yuan, Yuexin Wu, William W Cohen, and Russ R Salakhutdinov. Review networks for caption generation. *Advances in neural information processing systems*, 29, 2016.

[75] Zhongliang Yang, Yu-Jin Zhang, Sadaqat ur Rehman, and Yongfeng Huang. Image captioning with object detection and localization. *CoRR*, abs/1706.02430, 2017.

[76] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[77] Lu Yu, Malvina Nikandrou, Jiali Jin, and Verena Reiser. Quality-agnostic image captioning to safely assist people with vision impairment. *arXiv preprint arXiv:2304.14623*, 2023.

[78] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2021.