Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.038376
Article

Enhanced Image Captioning Using Features Concatenation and Efficient Pre-Trained Word Embedding

Samar Elbedwehy[1,3,*], T. Medhat[2], Taher Hamza[3] and Mohammed F. Alrahmawy[3]

[1]Department of Data Science, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh, 33511, Egypt
[2]Department of Electrical Engineering, Faculty of Engineering, Kafrelsheikh University, Kafrelsheikh, 33511, Egypt
[3]Department of Computer Science, Faculty of Computer and Information Science, Mansoura University, Mansoura, 35516, Egypt
*Corresponding Author: Samar Elbedwehy. Email: samarelbedwehy@ai.kfs.edu.eg

Received: 10 December 2022; Accepted: 02 February 2023

Abstract: One of the issues in Computer Vision is the automatic development of descriptions for images, sometimes known as image captioning. Deep Learning techniques have made significant progress in this area. The typical architecture of image captioning systems consists mainly of an image feature extractor subsystem followed by a caption generation lingual subsystem. This paper aims to find optimized models for these two subsystems. For the image feature extraction subsystem, the research tested eight different concatenations of pairs of vision models to get among them the most expressive extracted feature vector of the image. For the caption generation lingual subsystem, this paper tested three different pre-trained language embedding models: Glove (Global Vectors for Word Representation), BERT (Bidirectional Encoder Representations from Transformers), and TaCL (Token-aware Contrastive Learning), to select from them the most accurate pre-trained language embedding model. Our experiments showed that building an image captioning system that uses a concatenation of the two Transformer based models SWIN (Shifted window) and PVT (Pyramid Vision Transformer) as an image feature extractor, combined with the TaCL language embedding model is the best result among the other combinations.

Keywords: Image captioning; word embedding; concatenation; transformer

1 Introduction

The internet and social media platforms have rapid growth, such as Flickr and Instagram, as the number of image files available online, has exploded. Many applications, e.g., content-based image retrieval, require automatically generating captions to these vast numbers of images. As a result, human intervention is no longer required. The methods for extracting features are computationally expensive, have high dimensionality, and are usually domain-specific. In recent years, the advancement in neural networks and processing power has increasingly pushed the research in developing automatic image captioning deep learning models. Image captioning is processing images; captioning them based on their contents. The generated caption must be able to describe the objects in the image with their relationship, the action they performed, and most importantly, describe them as simply as possible while maintaining accuracy and omitting unnecessary information. There is a lot of research on improving and optimizing image captioning models. This is accomplished by optimizing the outcomes. An image can have several captions, but the task is to choose the best one to describe it. NLP (Natural Language Processing) is used to accomplish this. A caption should include all of the objects in the image, as well as their relationships with one another and the action they represent; where the other challenge is a high variance that exists in image captioning that arises when deep learning models are used to learn the specifics of training data. Because humans can perceive the things in an image and their interactions; and connect them to generate valuable sentences, writing a short caption while seeing the image is quite simple. A computer is trained to learn from various images and gather experience in the same manner that humans do.

This research aims to optimize the image captioning systems by first using a concatenation of vision transformer models to enhance the feature extraction part of the image captioning process; then optimize the language partly by evaluating the use of three different pre-trained language embedding models, namely GloVe, BERT and TaCL to get the best model among them to generate the most accurate generated captions by comparing two different frameworks; the first is using individual vision models for the feature extraction then use the three different pre-trained word embedding layer in the decoder phase each time and the second one is to use the concatenation feature extraction from the best one from the first framework then use the three pre-trained word embedding each time to find the best combinations from this comparisons.

Our experiments showed that a concatenation of the two Transformer based models, SWIN and PVT, that used an image feature extractor, combined with the TaCL language embedding model, produced the best results among the other combinations.

This paper is organized as follows; Section 2 propose the related work for image captioning and the word embedding used in image captioning. Section 3 discusses the proposed framework. Section 4 presents the evaluation stage in two sub-sections, one for the three pre-trained word embedding models with eight models for the vision stage, by comparing the results of the eight vision models. The other sub-section for the concatenation vision models with others and also applied the three pre-trained word embedding and made a comparison for them to find the best one. Finally, Section 5 presents the conclusion and future work.

2 Related Work

A framework for learning a transformation from one representation to another is the encoder-decoder. According to this framework, an encoder network first converts the input into a context vector, decoded by a decoding network to produce the output. Recurrent neural networks (RNNs) were recently introduced for sequence-to-sequence learning with applications to machine translation, where the input is a text sequence in one language and the output is a text sequence in the other language. This technique is known as encoder-decoder learning. Gu et al. [1] introduced a language CNN model that excels in image captioning and is appropriate for statistical language modeling applications. The long-range dependencies in historical words, which are essential for image captioning, may be modeled by CNN because it is fed with all the last words. Lu et al. [2] suggested a brand-new encoder-decoder framework for adaptive attention that offers the decoder a backup strategy. Their model chooses whether to focus on the picture (and if so, which regions) or the visual sentinel at each time step. To gather pertinent information for the production of successive words, the model determines whether and where to focus on the image. Wang et al. [3] provided a coarse-to-fine method that separates the original image description into a skeleton sentence and its attributes (Skel-LSTM+Attr-LSTM) before generating the skeleton sentence and attributes phrases. Because the fact that RNNs and LSTMs cannot compute in parallel and do not take into account the sentence's underlying hierarchical structure. Convolutional neural networks (CNNs) are the only type of technology used in the framework that was proposed by Wang et al. [4]. Their basic model performs better than NIC (an LSTM-based model) and is around three times faster during training, thanks to parallel computing.

For image captioning, Aneja et al. [5] suggested a convolutional architecture. They employ a feed-forward network devoid of recurrent operations. The method's architecture consists of the following four parts: input embedding layer, image embedding layer, convolutional module, and output embedding layer are the components of the algorithm. To make use of spatial visual properties, it also employs an attention mechanism. They test their design on the complex MSCOCO dataset and find that it performs comparably to an LSTM-based approach in standard measures. Yang et al. [6] proposed Scene Graph Auto-Encoder (SGAE), which incorporates the inductive language bias into the encoder-decoder image captioning framework for more human-like captions. It is expected that using such bias as a language prior will help the conventional encoder-decoder models less likely to over-fit the dataset bias and focus on reasoning. Humans naturally employ inductive bias when crafting collocations and contextual inference in language.

Wang et al. [7] proposed the Hierarchical Attention Network (HAN), which permits simultaneous attention calculations on features arranged in a pyramidal hierarchy. The pyramidal hierarchy contains features on many semantic levels, allowing for the prediction of various words based on various features. On the other hand, a Multivariate Residual Module (MRM) is suggested to learn the joint representations from features due to the many modalities of features. The MRM can extract pertinent relationships between various features and model projections. To balance the contribution of different aspects, they added a context gate. Parikh et al. [8], combined the model of CNN and GRU to achieve accurate image captions. Since the training effectiveness and expression ability of CNN-LSTM-based architectures were constrained, researchers started investigating CNN-Transformer-based models and had significant results. To improve visual representations, Zhang et al. [9] proposed the Grid-Augmented (GA) module, which incorporates relative geometry characteristics between grids. To extract language context, they created a BERT-based language model. They then proposed an Adaptive-Attention (AA) module on top of a transformer decoder to adaptively quantify the contribution of visual and linguistic signals before generating word prediction decisions. They built Relationship-Sensitive Transformer (RSTNet) for the image captioning challenge by applying the two modules to the basic transformer model. Xu et al. [10] suggested a cutting-edge Anchor-Captioner technique. To be more precise, they started by locating the significant tokens that should be given greater attention and treated as anchors. Then, they organized the relevant phrases for each selected anchor to create the appropriate anchor-centered graph (ACG). Last but not least, they performed multi-view caption generation based on various ACGs to increase the content diversity of generated captions. Using their methodology, they produced numerous captions that precisely and thoroughly represent various aspects of an image. Wang et al. [11] constructed a model that is purely Transformer-based, incorporates picture captioning into a single step, and enables end-to-end training. To extract grid-level features from provided pictures, they adopted Swin-Transformer to replace Faster R-CNN as the backbone encoder; additionally, the decoder converts the sophisticated features into captions word by word.

To increase the contribution of visual information for accurate prediction, Wu et al. [12] presented a Dual Information Flow Network (DIFNet), which uses the segmentation feature as an additional source of visual information to augment grid characteristics. They simply need a straightforward fusion approach because it is simple to combine grid characteristics and segmentation features. They suggested an efficient feature fusion module called Iterative Independent Layer Normalization (IILN), which can condense the most pertinent inputs by a standard LN layer while retraining modality-specific information in each flow via private LN layer, to maximize the benefits of two visual information flows. By comparing the usage of four distinct vision transformer models for the vision sub-models of the image captioning process, Elbedwehy et al. [13] evaluated the impact of employing the vision transformers on the image captioning process. DINO was the first vision transformer used (self-distillation with no labels). The second is a vision transformer called PVT, which does not employ convolutional layers. The third technique is XCIT (Cross-Covariance Image Transformer), which modifies the action of self-attention by concentrating on feature dimensions rather than token dimensions. The final one is SWIN, a vision transformer that, in contrast to the previous transformers, splits the image using shifted windows. The findings demonstrate that, in comparison to previous models, the suggested image captioning model's use of the SWIN transformer is highly effective. The most popular benchmark for image captioning used, is the MS COCO (Microsoft Common Objects in Context) dataset [14], as shown in Table 1, summarizes the related works which mentioned in this section.

Table 1: Previous work for encoder-decoder Image captioning

Researcher	Year	Encoder	Decoder	Data sets
Gu et al. [1]	2017	VGGNet	CNN but adding multimodal to connect encoder with decoder	MS COCO and Flickr30k
Lu et al. [2]	2017	ResNet	LSTM	MS COCO and Flickr30k
Wang et al. [3]	2017	CNN	Skel-LSTM+Attr-LSTM	MS COCO and a larger scale Stock3M
Wang et al. [4]	2018	CNN	CNN	MS COCO
Aneja et al. [5]	2018	VGGNet	Language CNN	MS COCO
Yang et al. [6]	2019	CNN	SGAE	MS COCO
Wang et al. [7]	2019	Faster-CNN	LSTM	MS COCO
Parikh et al. [8]	2020	CNN	Glove embedding+GRU	MS COCO 2017
Zhang et al. [9]	2021	RSTNet	Pre-trained BERT-based language	MS COCO
Xu et al. [10]	2021	DIFNet	Transformer Decoder	TextCaps dataset
Wang et al. [11]	2022	SWIN	LSTM	MS COCO
Wu et al. [12]	2022	Faster-CNN	BERT-BASE	MS COCO
Elbedwehy et al. [13]	2022	Transformer models	LSTM	MS COCO

Word embedding, on the other hand, is another enhancement that can be done to enhance the image captioning model specific to the language decoder. It has been employed as a distinct embedding layer for Image Captioning, which accepts a sequence of words as input and converts them to numbers matching their place in the vocabulary, then returns a vector of length n, where n is the embedding length. The weights of this embedding layer are learned by back-propagation, or they are adopted from another language modeling. There are few researchers on this point, such as Quanzeng et al. [15], who used Pennington et al. [16] pre-trained Glove word vectors to encode word information into the LSTM (Long Short-Term Memory). Embedding of Glove enhances Mikolov et al. [17] skip-gram model, but both rely on the text's linear sequential properties. To put it another way, both use the text's word co-occurrence properties to generate representations of the text. Other studies, such as Vinyals et al. [18], have found that using pre-trained embedding does not improve model performance. Atliha et al. [19] used pre-trained embedding, which is GloVe, with fine-tuning and found that it can improve the performance of the model and are most suitable for the image captioning model training improvement.

3 The Proposed Framework

Automated text captioning is a difficult task to accomplish; hence, it is usually built using a complex architectural model. In our proposed model for image captioning, there are two sub-models for the task of image captioning; a vision-model stage that acts as a vision encoder, extracting features from input images using a computer vision model, and a language model that acts as a decoder, converting the features and objects provided by the image sub-model into natural sentences.

Our goal in this paper is to optimize the proposed image captioning architectural model by comparing two different frameworks. The first is to optimize the vision encoder by using the best vision model for extracting features from images (encoder), then to get the most accurate generated captions by optimizing the language decoder models with three different pre-trained embeddings: GloVe, BERT, and TaCL to get the best combination. The research conducted a set of existing transformer-based vision models to find the best vision encoder model among them. The second framework in these experiments is concatenating the best pairs of the visual models that were built in the first framework and then applying the language decoder models with the three different pre-trained embeddings: GloVe, BERT, and TaCL to get the best one.

The comparison will be between these two frameworks to choose the best one for the image captioning system. As in Fig. 1, choose several images for training and testing, and then make processing the images before extracting the features by converting them to gray-scale and resizing the original image, and then extracting the features for the image using the vision models. Finally, the features are used as an encoder to the language decoder to generate the caption. The second framework is the exact process as in Fig. 1, but the difference is the concatenated vision model in the extracting feature phase with the same pre-trained word embedding.

4 Results and Discussion

The main goal of the evaluation process in this paper is to find the best combination of a vision encoder and a language decoder model to build a high-performance image captioning model. The vision decoder models tested in the evaluation include eight individual and six concatenated models. Each of these models tested an encoder that extracts features from images. Also, the paper used three different pre-trained embedding models in the language phase to choose the most accurate language decoder for our image captioning model. So, the research conducted the following individual experiments:

- Experiments for evaluating the individual image encoders with pre-trained Language embedding models in the language decoder.
- Experiments for the evaluation of the encoders using feature concatenation models with pre-trained Language embedding models in the language decoder.

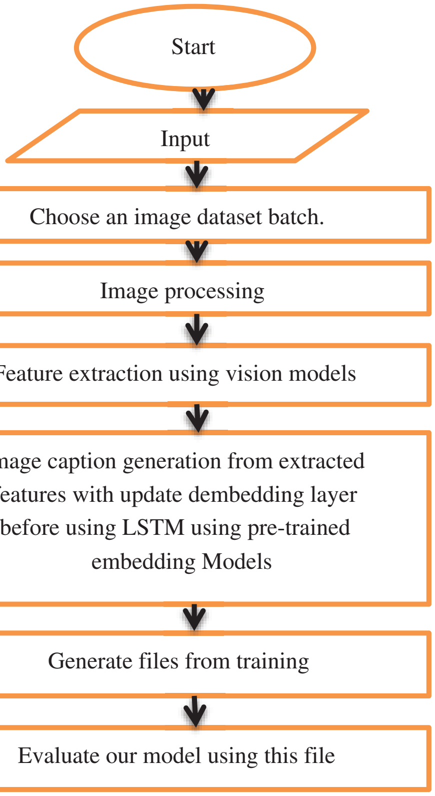

Start
Input
Choose an image dataset batch.
Image processing
Feature extraction using vision models
Image caption generation from extracted features with update dembedding layer before using LSTM using pre-trained embedding Models
Generate files from training
Evaluate our model using this file

Figure 1: The proposed pre-trained embedding for image captioning

This research used MS COCO as the framework for the evaluation of the proposed method. To be consistent with previous work, the paper used 30.000 images for training and 5000 images for testing and trained our model in an end-to-end using Keras model using a laptop with one GPU (2060 RTX). The hyper-parameter settings for all experimental models are as follows: Maximum Epochs are 30, LSTM dropout settings are [0.5], the learning rate is [4e-4], Optimizer: Adam optimizer; and finally, the batch size is 16. The details of each of the experiments are presented next. Fig. 2, shows the Plot of the Caption Generation Deep Learning Model for using SWIN with the PVT model, where input1 is the input of image features; input2 is the text sequences or captions. Dense is a vector of 2048 elements processed by a dense layer to produce a 256-element representation of the image as all the settings are the same in eight models used in this paper with their different methods, except the shape of the image will be changed, upon the concatenation model shape. With used the netron site [20] to plot the model by uploading the file of the model.

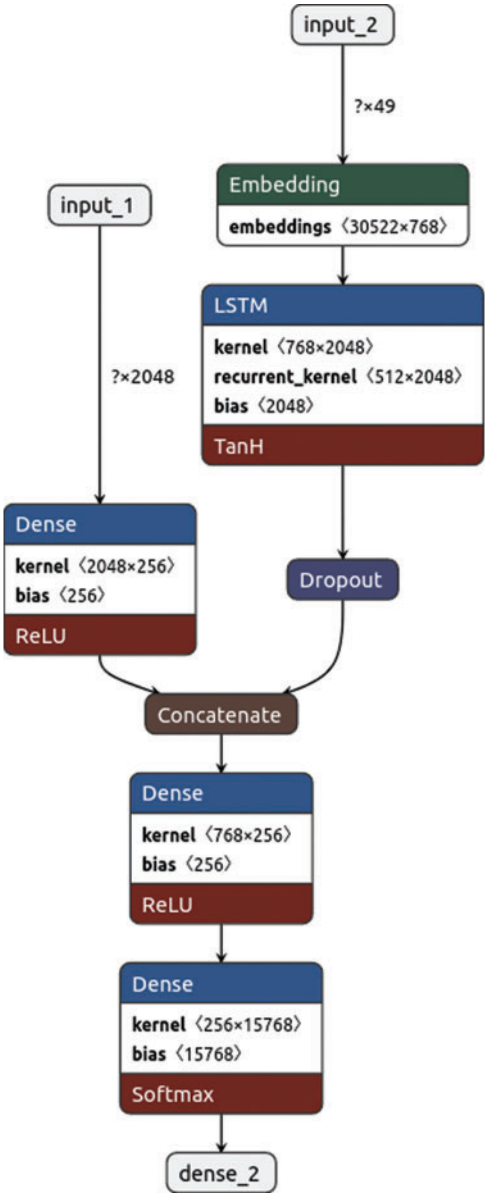


Figure 2: Plot of the caption generation deep learning model for SWIN with PVT

4.1 Experiments for the Evaluation of the Individual Vision Decoders with Pre-Trained Language Embedding Models

In these experiments, six different transformer-based models have been tested as image encoder models. These image encoder models are VGG16 [21], ResNet50 [22], ViT [23], and DINO transformer [24] that has been used with different backbones, including ResNet50, ViT s/8, ViT s/16, Xcit_meduim_24/p8, ViT b/8, the PVT_v2_b5 version of PVT [25], XCIT-Large version of the XCIT model [26] and finally, SWIN-Large version of SWIN-transformer presented in [27]. Each of the above image encoders has been tested in a separate experiment three times; each time, the tested encoder model combined with one of three different pre-trained embedding decoder models that are tested in our proposed model. These pre-trained embedding models are Glove [16], BERT [28], and TaCL [29]. The paper applied these decoder models in the language stage by using a fixed language decoder which is the LSTM-based model [30], which uses the feature vectors obtained from the proceeding tested vision models to generate the captions. Pre-trained embedding models are tested after extracting features from different vision models to compare the accuracy. The results of these experiments are presented and discussed next.

4.1.1 The Efficiency of Image Captioning

To show how the image captioning model is effective for creating captions for the input image, the paper utilized a popular metric for this criterion which is bilingual evaluation understudy BLEU (**Bil**ingual **E**valuation **U**nderstudy)-1, 2, 3, and 4. It is a well-liked statistic for measuring the effectiveness of MT (Machine Translation). BLEU scores [31] compare translated content to one or more reference translations. The evaluation process is going on by evaluating each generated caption to all of the image's reference captions and determining that it was pretty famous for captioning tasks. For cumulative n-grams, BLEU scores for 1, 2, 3, and 4 are determined. More particularly, BLEU1 and BLEU4 variants have been employed in image captioning techniques. Concerning the references, it calculates an n-gram-based precision for the candidate sentence. The central concept of BLEU is computing precision through clipping. Using most instances of a word in any reference phrase, clipping determines a word's accuracy. Therefore, if the word "The" appears no more than once in each reference, the candidate sentence "The The The" would receive credit for only using one "The". To discourage overly brief sentences, BLEU calculates the geometric mean of the n-gram precisions and applies a brevity penalty. Lower-order versions, such as BLEU1 (unigram BLEU) and BLEU2 (unigram and bigram BLEU), are also used. The research computes BLEU at the sentence level for evaluating image captioning. BLEU is most frequently calculated for machine translation at the corpus level, with a high correlation with human assessment; the association is weak at the level of individual sentences [31]. In this paper, there are particularly concerned about assessing caption accuracy. Table 1 shows the final results of comparing all models on all metrics on the validation dataset. It demonstrates that using pre-trained TaCL with most image encoder models is the best option. In the case of using the VGG-16 model, the measured values for BLEU-1 to 3 are better for the BERT model, which may indicate that the BERT model may work better in this case; however, as the value of BLUE-4 score in the case of using TaCL is better than the same metric when using BERT, this confirms that TaCL provides better results than BERT as with the other evaluated vision models, then this is because researchers consider BLEU-4 is more accurate and trusted more than the other scores and that is why BLEU-4 is the most popular BLEU formulation. Similarly, for the XCIT model, BLEU scores 1, 2, and 3 are also slightly higher than TaCL, but BLEU-4 for TaCL is higher than BERT. That means that the pre-trained TaCL embedding improves the BLEU scores, i.e., it enhances the efficiency of the image captioning process when compared to a model with BERT and GloVe, as shown in Table 2. Also, combining the TaCL language decoder with the SWIN transformer encoder produces the best captioning results. Fig. 3, shows a visual comparison of the BLEU scores results scored in these experiments. It shows that the SWIN model with TaCL pre-trained word embedding is the best one in the BLEU scores and the VGG-16 with GloVe pre-trained word embedding is the worst one.

Table 2: Image captioning efficiency measurements comparisons on MSCOCO [14]. All models are fine-tuned with self-critical training

Model-name	Word pre-trained model	BLEU1	BLEU2	BLEU3	BLEU4
VGG-16	GloVe	0.182691	0.054946	0.029173	0.009253
	BERT	**0.422661**	**0.143915**	**0.074355**	0.024252
	TaCL	0.369660	0.123343	0.074084	**0.026229**
ResNet50	GloVe	0.478455	0.309505	0.228672	0.126406
	BERT	0.631962	0.404957	0.292831	0.155725
	TaCL	**0.649572**	**0.419829**	**0.309214**	**0.172534**
ViT	GloVe	0.481062	0.311615	0.231212	0.128370
	BERT	0.636027	0.415404	0.303624	0.165749
	TaCL	**0.657128**	**0.428972**	**0.317406**	**0.178793**
PVT_v2_b5	GloVe	0.491928	0.325042	0.239991	0.133762
	BERT	0.657305	0.436736	0.318670	0.175278
	TaCL	**0.668987**	**0.444167**	**0.327661**	**0.183247**
DINO-ViTb/8	GloVe	0.499972	0.332085	0.247125	0.137330
	BERT	0.651690	0.434092	0.315040	0.169494
	TaCL	**0.672443**	**0.450144**	**0.335233**	**0.191341**
DINO-xcit_medium_24_p/8	GloVe	0.497019	0.330176	0.247560	0.140871
	BERT	0.642491	0.426253	0.310697	0.168561
	TaCL	**0.668875**	**0.446479**	**0.332452**	**0.188290**
XCIT	GloVe	0.501611	0.330679	0.245287	0.136811
	BERT	**0.669751**	**0.450073**	**0.329605**	0.181437
	TaCL	0.668808	0.444735	0.329494	**0.185424**
Swin-transformer	GloVe	0.527454	0.358555	0.269499	0.156086
	BERT	0.694754	**0.480268**	0.354753	0.200634
	TaCL	**0.696249**	0.480251	**0.361406**	**0.210658**

4.1.2 Time Evaluation

For each of the three tested pre-trained embedding, the research compared the time taken for training for each model to get the best epoch for captioning. As shown in Fig. 4, the DINO model with GloVe pre-trained embedding was the fastest in training as it took the least training time (2.18 h), and PVT with TaCL and BERT was the slowest, as it finished training in 12.8 h while SWIN, which is the most efficient in producing caption, has taken 10.9 h.

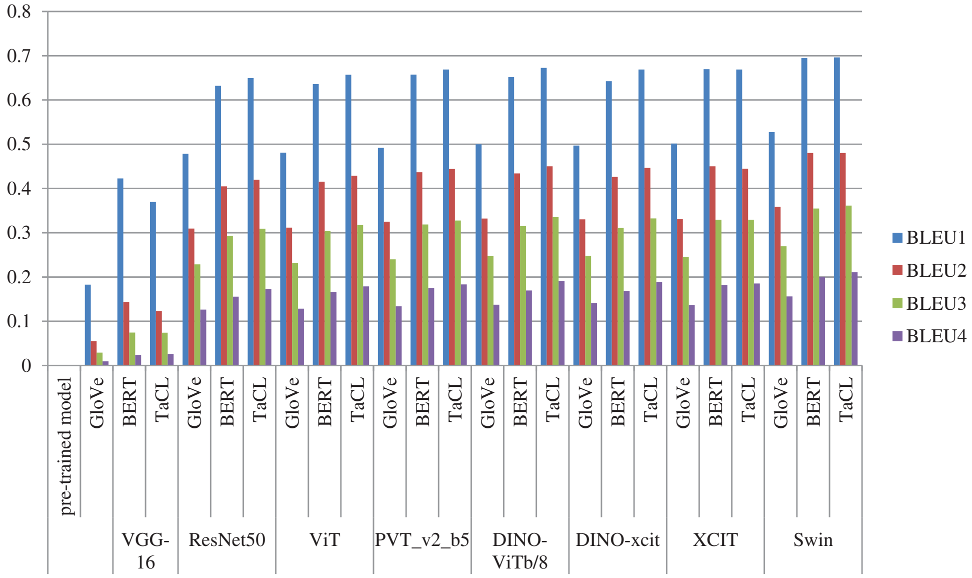


Figure 3: Comparisons between BLEU scores for GloVe, BERT, and TaCL

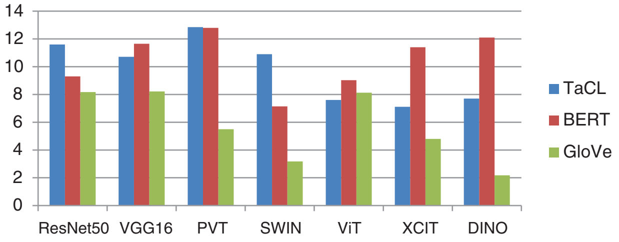


Figure 4: Training times of the tested image captioning models

4.1.3 *Performance Evaluation*

The performance evaluation is an essential criterion as it reflects how fast the image captioning model in generating captions of an input image. The paper used the Flops metric to evaluate the performance for the eight tested image captioning models with the three pre-trained embeddings (GloVe, BERT, and TaCL), as FLOPs (Floating Point Operations Per Second) are used to describe how many operations are required to run a single instance of a given model. The more FLOPs, the more time model will take for inference, i.e., the better models have a smaller FLOPS. Table 3 shows the number of FLOPS of each of the eight tested image captioning models. The worst model, PVTv2-b5 with BERT and TaCL, was the worst, while VGG and ViT with GloVe were the fastest models in generating the captions, while the SWIN model was a bit slower.

4.1.4 *Visual Evaluation*

Different samples of the image captioning produced by the tested models are shown in Fig. 5; on the left are the given images, and on the right are the corresponding captions. The captions in each box are from the same model sample. The research shows the captions from all the tested models. Captioning sentences with TaCL is more accurate than using the GloVe and BERT pre-trained embedding, especially with using the features of the SWIN-transformer model.

Table 3: Number of FLOPS for the tested Image captioning models

Captioning model using	Flops (TaCL)	Flops (BERT)	Flops (GloVe)
VGG16	32.52	32.52	**5.67**
ResNet50	32.59	32.59	5.73
ViT	32.52	32.52	**5.67**
PVTv2-b5	**33.44**	**33.44**	6.59
DINO-ViTb8	32.92	32.92	6.06
DINO-XCIT-m24_p8	32.79	32.79	5.93
XCIT	32.65	32.65	5.8
SWIN	32.59	32.59	5.73

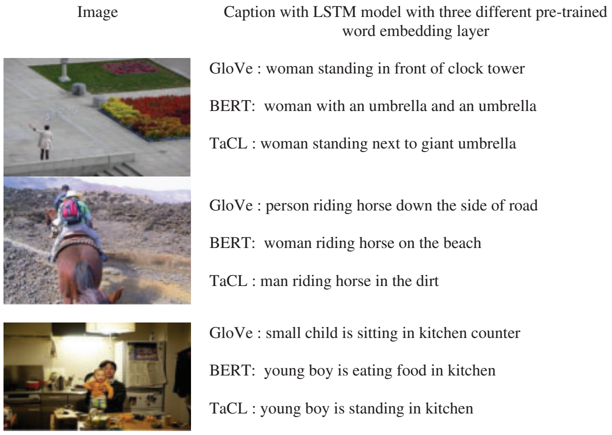


Image	Caption with LSTM model with three different pre-trained word embedding layer
	GloVe : woman standing in front of clock tower BERT: woman with an umbrella and an umbrella TaCL : woman standing next to giant umbrella
	GloVe : person riding horse down the side of road BERT: woman riding horse on the beach TaCL : man riding horse in the dirt
	GloVe : small child is sitting in kitchen counter BERT: young boy is eating food in kitchen TaCL : young boy is standing in kitchen

Figure 5: Samples for comparison produced by the tested image captioning models

4.2 Experiments for the Evaluation of the Decoders Using Feature Concatenation Models

The research evaluates in these experiments the different vision encoder models using feature concatenations from the set of individual vision models tested in our proposed model. These new concatenated models are DINO-ViTb/8 with PVT_v2_b5, SWIN with DINO-ViTb/8, SWIN with PVT_v2_b5, SWIN with PVT_v2_b5 with DINO-ViTb/8, XCIT with PVT_v2_b5 and finally SWIN with XCIT. The paper build these concatenated versions from the most efficient vision encoder models, as seen from the results of the experiments made using individual image encoder models. In each experiment, done in the last experiments, the concatenated model is used to extract features from images. These extracted features are fed into one of the three tested pre-trained embedding decoders (TaCL GLoVe, BERT). The research used the previous criteria defined earlier for evaluating the use of different feature concatenations for the image captioning model. The evaluation details are given next.

4.2.1 Efficiency of Image Captioning

Table 3 shows the final BLEU score values of the experiments using each of these tested concatenated image encoder models with one of the three pre-trained embedding decoders, as presented earlier. It demonstrates that using the feature concatenation encoder model (SWIN+PVT) with the

TaCL pre-trained embedding language decoder is the best combination as it improves the BLEU score compared to other concatenated models, as shown in Table 4. Fig. 6 shows the comparison of BLEU scores for using the tested models. Also, using the TaCL decoder always produces BLEU scores for all models very close to the highest scores. This sign indicates the high efficiency of this decoder model. It shows that SWIN+PVT model with TaCL pre-trained word embedding is the best one in the BLEU scores and the XCIT+PVT with GloVe pre-trained word embedding is the worst one.

Table 4: Image captioning efficiency measurements comparisons on MSCOCO [14]. All models are fine-tuned with self-critical training

Model-name	Word pre-trained model	BLEU1	BLEU2	BLEU3	BLEU4
Pvt+dino	GloVe	0.502683	0.335205	0.250328	0.141503
	BERT	0.649230	0.435131	0.316925	0.171032
	TaCL	0.678911	0.458720	0.341976	0.195557
Swin+dino	GloVe	0.521194	0.351736	0.263730	0.151466
	BERT	0.667640	0.455475	0.334166	0.183498
	TaCL	0.688394	0.472511	0.355310	0.205986
Swin+pvt	GloVe	0.524827	0.358543	0.269492	0.155086
	BERT	0.690018	0.476092	0.353077	0.200935
	TaCL	**0.697349**	**0.481288**	**0.363971**	**0.214781**
Swin+pvt+dino	GloVe	0.519266	0.350606	0.261292	0.148521
	BERT	0.673541	0.457955	0.334677	0.182340
	TaCL	0.688323	0.473250	0.354892	0.205088
Xcit+pvt	GloVe	0.503047	0.334039	0.249168	0.140514
	BERT	0.670101	0.451968	0.328810	0.179885
	TaCL	0.680049	0.459726	0.341656	0.194148
Swin+xcit	GloVe	0.523473	0.356581	0.267723	0.153658
	BERT	0.675592	0.465240	0.344109	0.192652
	TaCL	0.692381	0.476984	0.357083	0.206931

4.2.2 Time Evaluation

For each of the conducted experiments, the paper compared the time taken for training each model to get the best epoch for captioning. As shown in Fig. 7, the SWIN+XCIT model with GloVe pre-trained embeddings was the fastest in training best-conducted model took the least training time (3.6 h), while PVT+DINO with TaCL was the slowest, as it finished training in 14.2 h while SWIN+PVT, which is the most efficient in producing caption, has taken 10.7 h.

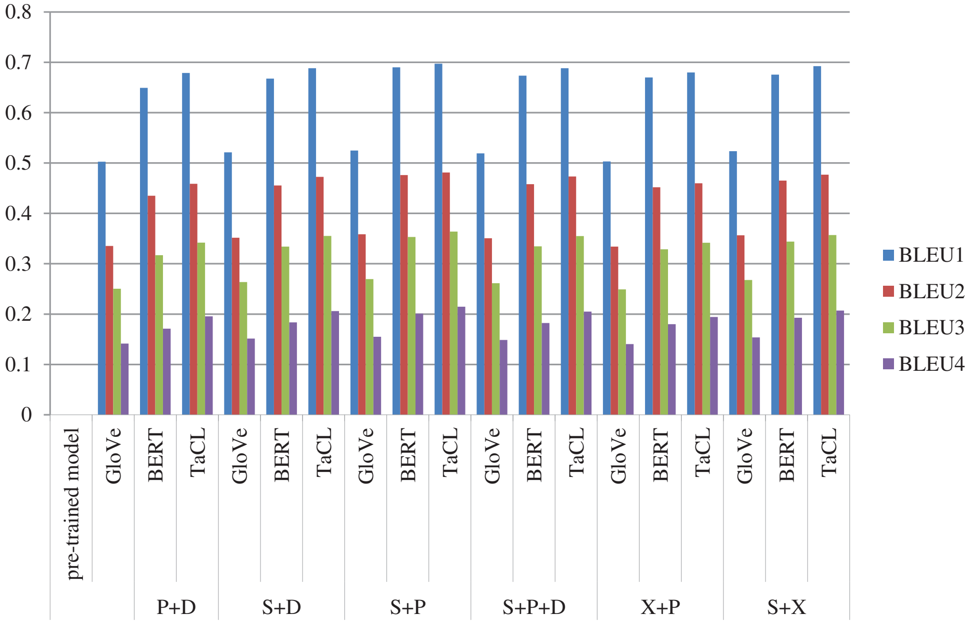


Figure 6: Comparisons between BLEU score for GloVe, BERT, and TaCL using the features concatenating models

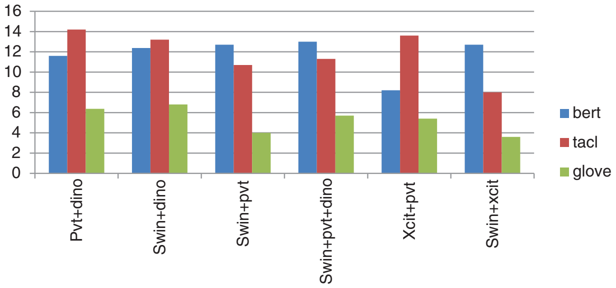


Figure 7: Training times of the tested image captioning models

4.2.3 Performance Evaluation

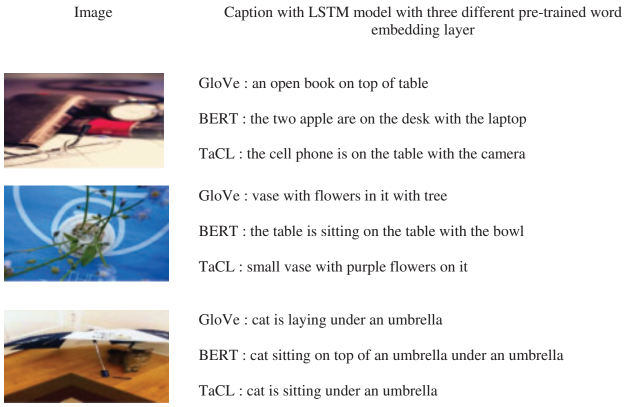
The paper measure the performance to evaluate how fast the image captioning model is in generating captions for an input image. The research used for this evaluation of the Flops metric, presented earlier, to evaluate the performance for the six concatenated tested captioning image models with the three pre-trained embeddings (GloVe, BERT, and TaCL); the better models should get a smaller number of FLOPS. Table 5 shows the FLOPS of each of the six concatenated tested captioning image models with the three tested decoders. All over the experiments, the worst model was SWIN+PVT+DINO, BERT, and TaCL, while the XCIT+PVT model with GloVe was the fastest model in generating the captions, while the SWIN+PVT model was a bit slower. Also, the SWIN+PVT+DINO model produces the worst performance with each of the used language decoders, while the XCIT+PVT model always produces the best performance.

Table 5: Number of FLOPS for the tested Image captioning models

Captioning model using	Flops (TaCL)	Flops (BERT)	Flops (GloVe)
Pvt+dino	32.85 m	32.85 m	5.99 m
Swin+dino	32.98 m	32.98 m	6.13 m
Swin+pvt	32.92 m	32.92 m	6.06 m
Swin+pvt+dino	**33.11 m**	**33.11 m**	6.26 m
Xcit+pvt	32.72 m	32.72 m	**5.86 m**
Swin+xcit	32.98 m	32.98 m	6.13 m

4.2.4 Visual Evaluation

Different samples of the image captioning produced by the tested models are shown in Fig. 8; on the left are the given images, and on the right are the corresponding captions. The captions in each box are from the same model sample. The paper shows the captions from all the tested models. Captioning sentences with TaCL is more accurate than using the GloVe and BERT pre-trained embeddings, especially with using the concatenation features by SWIN-transformer with the PVT model.



Image

Caption with LSTM model with three different pre-trained word embedding layer

GloVe : an open book on top of table

BERT : the two apple are on the desk with the laptop

TaCL : the cell phone is on the table with the camera

GloVe : vase with flowers in it with tree

BERT : the table is sitting on the table with the bowl

TaCL : small vase with purple flowers on it

GloVe : cat is laying under an umbrella

BERT : cat sitting on top of an umbrella under an umbrella

TaCL : cat is sitting under an umbrella

Figure 8: Samples for comparison produced by the tested image captioning models

5 Conclusion and Future Work

The paper focused on obtaining the best (image encoder-language decoder) integration to build a highly efficient Image Captioning model. It evaluated five different image captioning models (combinations of 14 different image encoders and three different language decoders). The best accuracy model in the first framework is SWIN+TaCL model, with 10.9 h in the training phase, but the best result is in the captioning sentences, while most of the models take more time for training and the little one comes with less accuracy. Also, in the second framework, the SWIN+PVT+TaCL model with 10.7 h is a little better than the first framework in terms of training time and in the accuracy results due to the feature concatenation that extracts the better features from the images than using the individual one. Hence, as concluded from the results, building an image captioning model that uses SWIN+PVT as an image encoder and TaCL as a language decoder can be considered an optimized architectural model for image captioning with relatively high accuracy. The paper aims in the future to use this architecture for producing Arabic captions of images, as research on generating Arabic descriptions of an image is extremely limited. Arabic has many challenging characteristics to learn, including writing from right to left, having many letters that are not pronounced by many other languages, and having more related words than English. Also, it aims to use it in applications such as image retrieval systems and Web mining.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Gu, G. Wang, J. Cai and T. Chen, “An empirical study of language CNN for image captioning,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 1222–1231, 2017.

[2] J. Lu, C. Xiong, D. Parikh and R. Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 375–383, 2017.

[3] Y. Wang, Z. Lin, X. Shen, S. Cohen and G. W. Cottrell, “Skeleton key: Image captioning by skeleton-attribute decomposition,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 7272–7281, 2017.

[4] Q. Wang and A. B. Chan, “CNN+CNN: Convolutional decoders for image captioning,” 2018. [Online]. Available: https://arxiv.org/abs/1805.09019

[5] J. Aneja, A. Deshpande and A. G. Schwing, “Convolutional image captioning,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 5561–5570, 2018.

[6] X. Yang, K. Tang, H. Zhang and J. Cai, “Auto-encoding scene graphs for image captioning,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 10685–10694, 2019.

[7] W. Wang, Z. Chen and H. Hu, “Hierarchical attention network for image captioning,” *Proceedings of the AAAI*, vol. 33, no. 1, pp. 8957–8964, 2019.

[8] H. Parikh, H. Sawant, B. Parmar, R. Shah, S. Chapaneri *et al.,* “Encoder-decoder architecture for image caption generation,” in *2020 3rd Int. Conf. on Communication System, Computing and IT Applications (CSCITA)*, Mumbai, India, IEEE, pp. 174–179, 2020.

[9] X. Zhang, X. Sun, Y. Luo, J. Ji, Y. Zhou *et al.,* “RSTNet: Captioning with adaptive attention on visual and non-visual words,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 15465–15474, 2021.

[10] G. Xu, S. Niu, M. Tan, Y. Luo, Q. Du *et al.,* “Towards accurate text-based image captioning with content diversity exploration,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 12637–12646, 2021.

[11] Y. Wang, J. Xu and Y. Sun, “End-to-end transformer based model for image captioning,” 2022. [Online]. Available: https://arxiv.org/abs/2203.15350

[12] M. Wu, X. Zhang, X. Sun, Y. Zhou, C. Chen *et al.,* “DIFNet: Boosting visual information flow for image captioning,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, pp. 18020–18029, 2022.

[13] S. Elbedwehy, T. Medhat, T. Hamza and M. F. Alrahmawy, “Efficient image captioning based on vision transformer models,” *Computers, Materials & Continua*, vol. 73, no. 1, pp. 1483–1500, 2022.

[14] M. Pedersoli, T. Lucas, C. Schmid and J. Verbeek, “Areas of attention for image captioning,” in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 1242–1250, 2017.

[15] Y. Quanzeng, J. Hailin, W. Zhaowen, F. Chen and L. Jiebo, "Image captioning with semantic attention," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 4651–4659, 2016.

[16] J. Pennington, R. Socher and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543, 2014.

[17] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[18] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 652–663, 2016.

[19] V. Atliha and Š. Dmitrij, "Pretrained word embeddings for image captioning," in *2021 IEEE Open Conf. of Electrical, Electronic and Information Sciences (eStream)*, Vilnius, Lithuania, pp. 1–4, 2021.

[20] L. Roeder, "Netron, Visualizer for neural network, deep learning, and machine learning models," 2017. [Online]. Available: https://doi.org/10.5281/zenodo.5854962

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770–778, 2016.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai *et al.,* "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[24] M. Caron, H. Touvron, I. Misra, H. Jégou, M. Julien *et al.,* "Emerging properties in self-supervised vision transformers," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Montreal, QC, Canada, pp. 9650–9660, 2021.

[25] W. Wang, E. Xie, X. Li, D. Fan, K. Song *et al.,* "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Montreal, QC, Canada, pp. 568–578, 2021.

[26] A. Ali, H. Touvron, M. Caron, P. Bojanowski, M. Douze *et al.,* "XCiT: Cross-covariance image transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20014–20027, 2021.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei *et al.,* "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision*, Montreal, QC, Canada, pp. 10012–10022, 2021.

[28] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[29] Y. Su, F. Liu, Z. Meng, T. Lan, L. Shu *et al.,* "Tacl: Improving BERT pre-training with token-aware contrastive learning," 2021. [Online]. Available: https://arxiv.org/abs/2111.04198

[30] B. Tarján, G. Szaszák, T. Fegyó and P. Mihajlik., "Investigation on N-gram approximated RNNLMs for recognition of morphologically rich speech," in *Int. Conf. on Statistical Language and Speech Processing*, Ljubljana, Slovenia, pp. 223–234, 2019.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556