IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 23, 2022, accepted March 9, 2022, date of publication March 22, 2022, date of current version March 31, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3161428

Deep Learning Approaches Based on Transformer Architectures for Image Captioning Tasks

ROBERTO CASTRO[1], (Student Member, IEEE), ISRAEL PINEDA[1], (Member, IEEE), WANSU LIM[2], (Member, IEEE), AND MANUEL EUGENIO MOROCHO-CAYAMCELA[1], (Member, IEEE)

[1]Deep Learning for Autonomous Driving, Robotics, and Computer Vision Research Group (DeepARC Research), School of Mathematical and Computational Sciences, Yachay Tech University, Urcuquí 100119, Ecuador
[2]Future Communications and Systems Laboratory (FCSL), Department of Aeronautics, Mechanical and Electronic Convergence Engineering, Kumoh National Institute of Technology, Gumi-si, Gyeongbuk 39177, Republic of Korea

Corresponding authors: Wansu Lim (wansu.lim@kumoh.ac.kr) and Manuel Eugenio Morocho-Cayamcela (mmorocho@yachaytech.edu.ec)

This work was supported in part by the Ministry of SMEs and Start-ups, South Korea, under Grant S3010704; and in part by the National Research Foundation of Korea under Grant 2020R1A4A101777511 and Grant 2021R1I1A3056900.

ABSTRACT This paper focuses on *visual attention*, a state-of-the-art approach for image captioning tasks within the computer vision research area. We study the impact that different hyperparemeter configurations on an encoder-decoder visual attention architecture in terms of efficiency. Results show that the correct selection of both the cost function and the gradient-based optimizer can significantly impact the captioning results. Our system considers the cross-entropy, Kullback-Leibler divergence, mean squared error, and negative log-likelihood loss functions; the adaptive momentum (Adam), AdamW, RMSprop, stochastic gradient descent, and Adadelta optimizers. Experimentation shows that a combination of cross-entropy with Adam is the best alternative returning a Top-5 accuracy value of 73.092 and a BLEU-4 value of 20.10. Furthermore, a comparative analysis of alternative convolutional architectures demonstrated their performance as an encoder. Our results show that ResNext-101 stands out with a Top-5 accuracy of 73.128 and a BLEU-4 of 19.80; positioning itself as the best option when looking for the optimum captioning quality. However, MobileNetV3 proved to be a much more compact alternative with 2,971,952 parameters and 0.23 Giga fixed-point Multiply-Accumulate operations per Second (GMACS). Consequently, MobileNetV3 offers a competitive output quality at the cost of lower computational performance, supported by values of 19.50 and 72.928 for the BLEU-4 and Top-5 accuracy, respectively. Finally, when testing vision transformer (ViT), and data-efficient image transformer (DeiT) models to replace the convolutional component of the architecture, DeiT achieved an improvement over ViT, obtaining a value of 34.44 in the BLEU-4 metric.

INDEX TERMS Image captioning, visual attention, computer vision, supervised learning, artificial intelligence.

I. INTRODUCTION

Image captioning is a branch of computer vision whose main objective is the generation of accurate and organic text descriptions of any type of scenario portrayed in an image or frame [1]. Traditional approaches (i.e., before the neural network's era) tackled the image captioning problem using classical image processing methodologies that usually relied on the generation of templates together with object detection to produce the caption given an input image [2], [3]. Following a similar line to the use of image templates, the construction of pattern recognition systems has made a meritorious historical space in the resolution of computer vision tasks involving images, as in the case of content-based image retrieval problems [4]. Moreover, the incorporation of fuzzy logic was of great interest over time as it positioned itself as a popular method that maps labels from previously extracted features [5], [6]. As a consequence of the emerging techniques, joined to the usage of neural structures, visual attention has emerged as a high potential alternative, proposing to replicate human vision by enabling an emulation of attention by the neural network on the most relevant sections of an image [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

Several researchers have replicated the state-of-the-art implementation proposed by Xu *et al.* for further study [8]. The latter convolutional architecture can be broadly divided into two well-defined structures. On the one hand, a convolutional network, which takes as input the raw images to be processed, while it outputs a set of feature vectors, each of which represents a *D*-dimensional part of a section of the illustration. Thus, the decoding part of the model will be able to selectively focus on specific parts of the image by making use of subsets of the feature vectors. In addition, a long short-term memory (LSTM) network makes use of the previous output to generate a word at each time instant in dependence on a context vector, previously generated words, and the previous hidden state.

Modern artificial intelligence models provide promising results for the captioning problem. However, one of the remaining challenges is the optimization of hyperparameters which is far from trivial and remains a challenge for captioning and other applications [9].

In this paper, three experimental scenarios are examined with the *Show, Attend and Tell* architecture as the object of study. First, we conduct a study that serves as complementary content to our paper, seeking to leave tangible evidence that support the general configuration of the original contribution. Otherwise stated, alternatives that equal or exceed the performance obtained in the benchmark work. In order to achieve the previously mentioned objective, it was decided to study the performance impact of different model hyperparameters, conducting a comparative study to select the cost function that minimizes the training error over a certain number of epochs for our specific application, setting the optimizer as a fixed variable. Then, the same principle is applied to test different gradient-based optimizers with the cost function as an independent variable. As a second experiment, once the optimal configuration of hyperparameters was established, we sought to study the performance and computational requirements that various convolutional models can achieve by replacing the original encoder. And finally, to analyze the viability of recent models that leave aside the notion of convolutions, we tested the performance of architectures based on transformers, replacing the encoder component of the baseline original work.

In response to the uncertainties raised by the previously described experimental scenarios, the combination of cross-entropy loss and Adam optimizer was highlighted as the best hyperparameter configuration according to the Top-5 Accuracy, BLEU-4, and loss value metrics. By reusing this configuration for the following experiment, different decisions can be made depending on the final purpose of the researcher [10]. If the architecture with the best metrics concerning response quality is required, the convolutional models ResNet-152 and ResNeXt-101 provided the best results in the metrics used in the previous experimentation. On the other hand, looking for the alternative with the lowest computational demands, the MobileNet V3 model is the most attractive, decreasing the number of parameters, training time, and inference, together with the giga fixed-point multiply-accumulate operations per second (GMACs), without sacrificing the accuracy metrics considerably. Finally, as the last experimental scene, it was decided to dispense with the original encoder used by the benchmark architecture in order to decide for alternatives outside the convolutional principles. Two different transformer-based models, initially conceived for image classification tasks, were selected for this last examination. According to the corresponding results, an improvement of state of the art in terms of the BLEU-4 metric was obtained when using the Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) models. However, the best results were obtained when using the second of these couple of models, in conjunction with a training process consisting of an initial phase where only the decoder of the architecture is subjected to training, while as a second stage, the parameters that conform the last transformer encoder block are also optimized.

II. RELATED WORKS

According to the historical summary presented in Table 1, one of the pioneering research works incorporating an *attention* system is the one proposed by Larochelle & Hinton, based on a variant of the *restricted Boltzmann machine* (RBM) mainly used for digit classification. They used the benchmark MNIST dataset, where a limited set of pixels is provided from which the architecture collects both high- and low-resolution information about neighboring pixels [11]. Moving forward in the timeline, Bahdanau *et al.* reused the notion of attention applied to different convolutional architectures. In this case, a much more novel model such as an *encoder-decoder* makes use of a reduced but visible attention system to take into consideration certain parts of a sentence when performing the translation of a specific word [12]. The idea of taking advantage of the benefits offered by *recurrent architectures* was a common factor that persisted in later works, among which stand out research-oriented to digit classification such as that presented by Mnih *et al.* [13], and the one proposed by Ba *et al.* [14].

In order to substantiate the evolution within the area of image captioning, a brief historical review of relevant works is presented in Table 2. Throughout this summary, we can find contributions such as the one proposed by Kiros *et al.*, using a *multi-log bilinear model* for exploiting the characteristics of images to generate a biased version of this architecture [15]. Followed this research, the same author incorporated recurrent structures within an encoder-decoder model, a common factor among image captioning proposals. This fact is mainly due to the nature of human speech that is sought to be incorporated into the learning algorithm. Furthermore, authors such as Mao *et al.* [16], Vinyals *et al.* [17], and Donahue *et al.* [18] have reused this idea in their respective research efforts.

Finally, Table. 3 contains an excerpt from previous works that promote our hypothesis of incorporating a non-convolutional model within the proposed benchmark.

TABLE 1. Summary of visual attention related works.

Architecture	Data input	Cost Function	Optimizer	Performance metric	Reference
Multi-fixation Restricted Boltzmann Machine (RBM)	Images	Hybrid Cost Hybrid-Sequential Cost	SGD	Error rate and accuracy	Larochelle & Hinton (2010)
Encoder-Decoder	Source sentence of 1-of-K coded word vectors	N/A	SGD and Adadelta	BLEU.	Bahdanau et al. (2014)
Recurrent Neural Network	Images	Cross entropy and Reinforcement.	SGD with momentum	Error rate	Mnih et al. (2014)
Deep Recurrent Attention Model	Images	Log-Likelihood	SGD with the Nesterov momentum	Error rate	Ba et al. (2014).
Encoder-Decoder	Images and encoded captioning	Cross entropy	Adam	BLEU and METEOR	Xu et al. (2016).

TABLE 2. Summary of image captioning related works.

Architecture	Data input	Cost function	Optimizer	Performance metric	Reference
RNN	Image and sentence descriptions.	Log-likelihood calculated by perplexity plus a regularization term.	N/A	BLEU, Perplexity, Recall@K and Median rank.	Mao et al. (2014)
LSTM	Image passes through a CNN.	Sum of the negative log likelihood of the correct word at each step.	SGD	BLEU, METEOR, CIDER, Recall@k and Median rank.	Vinyals et al. (2014)
LSTM	Images or Text	Negative log likelihood	SGD	BLEU, METEOR, CIDER, Recall@k, Median rank and Rogue-L.	Donahue et al. (2014)
Multimodal log-bilinear model	Images	Perplexity	N/A	BLEU, Perplexity	Kiros et al. (2014a)
Encoder-Decoder	Images	Pairwise ranking loss	SGD	Recall@k and Median rank	Kiros et al. (2014b)

TABLE 3. Summary of related works about transformer architectures.

Architecture	Data Input	Task	Dataset	Cost Function	Optimizer	Performance Metric	Reference
Original Full Transformer	Text	Machine Translation	WMT 2014: - English-to-German - English-to-French	N/A	Adam	BLEU, FLOPS	Vaswani et al. (2017)
Transformer Encoder + RNN	Text	Machine Translation	- NIST OpenMT Chinese-to-English - WMT 2017 Chinese-to-English	- Fist Stage: Negative Log-Likelihood - Second Stage: Sequence-level Knowledge Distillation KD	N/A	BLEU	Wang et al. (2019)
CNN Encoder + Transformer Decoder	Images	Image Captioning	Flickr8k	N/A	Adam	BLEU, METEOR, CIDEr, ROUGE	Patel & Varier (2020)
Transformer encoder (BERT) + LSTM	Text	Sentence Correction	NLPCC 2018	N/A	Adam	BLEU	Chen et al. (2020)
Transformer Encoder + MLP	Images	Image Classification	- ImageNet - CIFAR 10/100 - Oxford-IIIT Pets/Flowers-102 - VTAB	N/A	Adam	Top-1 Accuracy	Dosovitskiy et al. (2021)
Full Vision Transformer	Images	Image Captioning	MS-COCO	Cross-entropy loss	Adam	BLEU, METEOR, CIDEr, ROUGE	Liu et al. (2021)

The transformer architecture originates with the proposition that *attentional systems* are sufficient tools to replace approaches that employ recurrent networks for machine translation tasks. This architecture uses *multi-head attention* as the cornerstone of the transformer blocks contained in the encoding and decoding part. The authors of this work use a simile with database information retrieval systems to propose its attentional principle, generating the key K, the query Q, and the value V matrices from linear projections on the input. This technique is intended to divide the aforementioned matrices for each attention head in order to compute the attention as follows:

$$Attention(Q, K, V) = \texttt{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

where d_k corresponds to the embedding size used to represent each word [19].

The achievement obtained in this work is evidenced by an improvement in the BLEU metric for English-to-German translation tasks compared to the state-of-the-art.

The novel transformer architecture attracted the attention of engineers and practitioners by dispensing the conventional convolutional or recurrent models, usually used to build encoders and decoders. Hence, researchers were fast to evaluate the feasibility of both parts that constructed this outstanding model.

On the one hand, regarding the machine translation tasks, the encoder of the transformer has been sought to be used as an alternative for the encoding of the content coming from an input text. One of the main attractions of this specific part of the transformer is the high parallelization capacity due to the nature of the multi-head attention modules. On the other hand, the decoder, similar to recurrent models, requires previous states when generating a new word during the inference process. Thus, Wang *et al.* proposed to counteract the impact of the large number of parameters of a transformer decoder by replacing it with a classical LSTM network to perform the translation task given the output generated by the transformer encoder. Thereby, the authors end up with an architecture capable of decoding four times faster than using the classical transformer, with a slightly lower performance in terms of BLEU metric [20].

As time went by, the scientific community became much more aware of the role that both transformer parts played in performing translation tasks. During training, the encoder acquires the general understanding of the source language, considering the context in which each word was initiated. At the same time, the decoder is trained to map the words from the source language to the target language. Therefore, the underlying knowledge of the language that both neural network architectures had separately granted to the scientific community, have provided two great weapons to tackle natural language tasks. On the one hand, by exploiting the decoder modules of the transformer we obtain the GPT architecture, whose later versions leave a hegemony mainly in text generation [21]. On the other hand, models such as BERT have been proposed to take advantage of the encoder modules. The versatility of this model is undeniable at the moment of performing almost any task in the area of natural language processing by executing fine-tuning according to the specific application [22].

Once the precedent set by BERT was established, its use in conjunction with recurrent networks continued to be a great experimental attraction thanks to the computational benefits mentioned above. Thus, Chen *et al.* proposed the acceleration of sentence correction tasks in Chinese, using a BERT-RNN model trained by applying the TF technique as an additional measure to accelerate the training process. After experimentation with various recurrent models functioning as decoder, the BERT-GRU combination outperformed the best BLEU metric, and improved the inference time of the base transformer model by 1131% [23].

Despite the progressive dominance of transformer-based networks in natural language processing, the feasibility of this type of architecture in the world of computer vision has been the focus of many researchers in the last couple of years. An example of the first approach to this new challenge can be found in the work of Patel and Varier. They contributed to the research community with a comparison between a CNN-LSTM model and a CNN-Transformer architecture for image captioning tasks on the *Flickr8k* dataset. This work concludes by showing the feasibility of the transformer decoder within the proposed architecture. However, the performance metrics remained slightly behind in terms of BLEU, METEOR, ROGUE and CIDER in comparison to the classical alternatives using LSTM networks as a decoder [24].

Subsequently, because of the considerable impact caused by the work "*An image is worth* 16×16 *words: transformers for image recognition at scale*" by Dosovitskiy *et al.*, the ViT model was considered as a viable approach to the use of transformer-based architectures for computer vision. The authors of this work proposed an architecture that uses the transformer encoder reusing configurations from the BERT model. The output of this encoder part is then reused within an multi-layer percepton (MLP) layer to perform image classification. The modification that allows this architecture to take an image as input, is that the corresponding input is previously divided into N patches, each one containing an specific section of the image, ensuring no overlapping between them. These image portions are then flattened and each of these structures is treated as if it were a word within the classical transformer architecture. The impact that this paper generated was not only due to the alternative proposed to use an image as input, but also for being a new state-of-the-art in the task of image classification [25].

After this recent approach of using transformers for tasks involving images had been consolidated, the desire to use a full-transformer architecture for this type of tasks continued to be studied. Liu *et al.* proposed the use of such an architecture, using the ViT model as the coding part together with the classical decoder of the transformer [26]. This proposal was tested in image captioning tasks on the

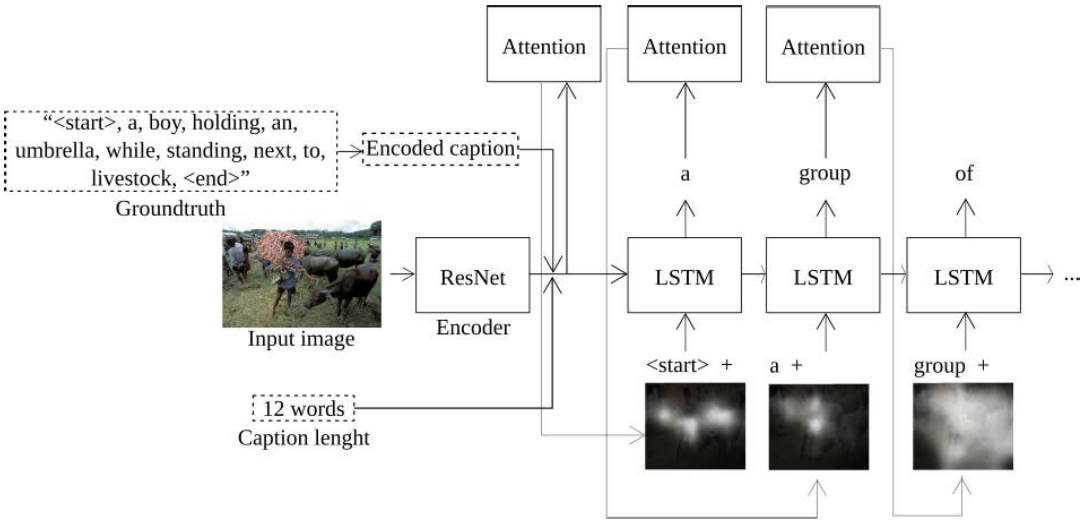


FIGURE 1. Overall representation of the convolutional encoder-decoder architecture built to generate real captioning.

MSCOCO dataset, obtaining an improvement of the state-of-the-art in terms of BLEU, METEOR, ROGUE and CIDER metrics.

As mentioned so far, the current trend corresponds to the exploitation of attentional systems based on transformers, even pursuing the possibility of consolidating a model capable of being specialized in multiple vision-language tasks after a short period of fine-tuning [27]. However, new approaches inspired by the one proposed in the *Show, Attend and Tell* work remain on the table as fierce competitors in the area of image captioning. Thus, progress continues to be made in the generation of descriptions in Chinese, using architectures that not only continue to employ convolutional structures for the extraction of features present in the images, but the decoding process remains in charge of a recurrent network, more specifically using bidirectional LSTM networks supported by a fuzzy attentional module [28].

III. SYSTEM MODEL AND DESIGN

The convolutional model employed for this study is built following an encoder-decoder architecture supported by a visual attention model. The proposed neural architecture is schematized in Fig. 1, where an instance of the dataset is outlined in order to show its operation. The encoder makes use of transfer learning by borrowing the original convolutional architecture of Resnet [29], taking the pre-trained model from the PyTorch repository.[1] This operation aims to generate an encoded version of the input RGB image composed by a set of L D-dimensional annotation/feature vectors, where each one corresponds to a simplified representation of a part of the original image.

$$a = \{a_1, a_2, \ldots, a_L\}, \quad a_L \in \mathbb{R}^D \tag{2}$$

[1]https://github.com/pytorch/vision/blob/main/torchvision/models/resnet.py

On the decoder side, given the sequential nature of the problem to be solved, an LSTM recursive architecture is constructed [30]. Up to this point, the description of the input image is generated in a word-by-word basis. At each decoding step, the *Att*-MLP attention network uses the set of annotation vectors together with the previous hidden state, passing this output through a softmax function.

$$\lambda_{ti} = Att(a_i, h_{t-1}) \tag{3}$$

$$\alpha_{ti} = \frac{exp(\lambda_{ti})}{\sum_{k=1}^{L} exp(\lambda_{tk})} \tag{4}$$

Once the corresponding weights have been computed for each annotation vector at time t, we proceed to compute the vector $\hat{z}_t$, which is a dynamic representation of the relevant parts of an image for an specific time. For the present work, we analyze the deterministic approach of the original architecture, parsing the context vector as a soft attention-weighted annotation vector.

$$\hat{z}_t = \sum_{i=1}^{L} \alpha_{ti} a_i \tag{5}$$

Through this outcome, the previously generated word and the previous hidden state, the LSTM network generates the corresponding output word probability:

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto exp(\mathbf{L}_0(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t)) \tag{6}$$

where $\mathbf{L}_0$, $\mathbf{L}_h$, $\mathbf{L}_z$, and $\mathbf{E}$ are learnable parameters initialized randomly. The objects in Fig. 1 denoted with discontinuous contours are the groundtruth components extracted from the dataset. Notwithstanding, those objects are only used during the training phase of the model. Their nature is described in the next section of the paper.

IV. THE DATASET STRUCTURE

The dataset used for training the network is the 2014 version of the MS COCO variant oriented to image captioning tasks [31]. Three inputs are structured in the dataset to be used by the neural network during the training stage. It should be noted that these three components are prepared for the training, testing, and validation sets.

A. INPUT IMAGES

The set of images obtained from MS COCO must have pixels values in the domain $b \in \{0, 1\}$ to be compatible with the pre-trained convolutional model used as the encoder block. For the effect, a normalization of the RGB channels is applied using the values of $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$, where μ and σ represent the mean and the standard deviation of the ImageNet dataset [32], respectively. Each image in the dataset is represented as $\boldsymbol{X}^{(i)} \in \mathbb{R}^{256\times256}$, where $\boldsymbol{X}^{(i)}$ is a matrix of 256×256 pixels. We let m be the total number of images on MS COCO dataset, and represent the entire dataset as $\boldsymbol{X} \triangleq \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(m)}\}$, where each image $\boldsymbol{X}^{(i)}$ is mapped to a ground truth caption $\boldsymbol{Y}^{(i)}$ that represents the corresponding ground-truth encoded caption.

B. ENCODED CAPTIONS

In order to be able to manipulate the descriptions associated with each image in the dataset, the model uses a mapping system supported by a dictionary. Within this file, each word used in the captioning of the entire dataset has an identification number. In this way, each ground-truth will be represented as a numerical array according to the equivalences defined by the mapping system.

In addition, the inclusion of three special characters within the mapping file is required. On the one hand, the neural network requires a *start* and *end* signal to delimit the extension of the descriptions. On the other hand, since not all the descriptions occupy the same sentence size, it is required to fill the missing spaces of the encoded caption with a padding character. Consequently, taking the longest ground-truth as referral, the content of the rest of the captions is updated to match the reference length by incorporating the padding operator. The proposed methodology normalizes the MS COCO dataset in arrays of 52 elements.

As an example, in Fig. 2 it can be seen an instance included in the validation group. This image is associated with a corresponding C description: "`a man with a red helmet on a small moped on a dirt road`". Referring to the file, which contains its encoded description E_C, one can find an encoding of the form:

$$E_C = [9488, 1, 2, 3, 1, 4, 5, 6, 1, 7, 8, 6, 1, 9, 10, 9489, 0, 0, \ldots, 0],$$

considering that it has been generated from the equivalences contained in the mapping file, the contents of which are presented in Table 4.

FIGURE 2. Image taken from the training set with an associated groundtruth caption: "`a man with a red helmet on a small moped on a dirt road`."

TABLE 4. Mapping system used to encode the caption the example image.

Word	Encoded Version
a	1
man	2
with	3
red	4
helmet	5
on	6
small	7
moped	8
dirt	9
road	10
...	...
<start>	9488
<end>	9489
<pad>	0

C. CAPTION LENGHTS

Finally, the last file is generated whose purpose is to house an array, whose elements represent the number of words that make up the description associated with each of the images.

V. HYPERPARAMETER NOTIONS

This section describes the loss and optimizer functions employed by the reference benchmark. In addition, Algorithm 1 details the intervention of these components during the training phase of the neural network.

A. CROSS-ENTROPY LOSS FUNCTION

To describe the loss function of our attention model, we let a be the function parametrized by $\boldsymbol{\theta}$, the caption output of the network is represented as $\boldsymbol{C} = a(\boldsymbol{X}, \boldsymbol{\theta})$, where $\boldsymbol{C}$ is the collection of words inferred from the MS COCO dictionary. The loss function measures the inference performance of our attention model when compared with its respective ground truth. In order to measure the difference between the ground truth distribution and the distribution of the caption outcome,

we define $J(\boldsymbol{\theta})$ as the *cross-entropy*. The cross-entropy loss function penalizes the attention model when it infers a low probability for a given caption. Our attention model works by updating the values of $\boldsymbol{\theta}$, moving the loss towards the minimum of $J(\boldsymbol{\theta})$ [33].

For our training set of $(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)})$ for $i \in \{1, \ldots, m\}$, we estimate the parameters $\boldsymbol{\theta} = \{\theta^{(1)}, \ldots, \theta^{(n)}\}$ that minimizes $J(\boldsymbol{\theta})$ by computing:

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{m}\sum_{i=1}^{m} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= -\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{Y}^{(i)} log\Big(\hat{p}^{(i)}\Big), \end{aligned} \tag{7}$$

where $\boldsymbol{Y}^{(i)}$ represents the expected caption $\mathbf{C}$ of the i^{th} image, and $\hat{p}^{(i)}$ constitutes the probability that the i^{th} image outcomes the intended value of $\mathbf{C}$.

B. ADAPTIVE MOMENT OPTIMIZER

In order to optimize our attention model through a gradient-based optimization method, we express the gradient vector of (7) with respect to θ as

$$\begin{aligned} \mathbf{g} &= \nabla_{\theta} J(\boldsymbol{\theta}) \\ &= \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= \frac{1}{m}\sum_{i=1}^{m} \Big(\hat{p}^{(i)} - \boldsymbol{Y}^{(i)}\Big)\boldsymbol{X}^{(i)}. \end{aligned} \tag{8}$$

To locate the minimum of $J(\boldsymbol{\theta})$, the proposed optimization algorithm moves to the negative direction of (8) iteratively. Our model computes individual adaptive learning rates for different parameters from estimates of first and second moments of $\mathbf{g}$ [34].

VI. EXPERIMENTAL SETTINGS

It is essential to point out that for the three study cases, the training of the corresponding models was performed considering that the aim was to take advantage of the use of transfer learning on the encoder part. Therefore, only the part of the architecture directly in charge of generating the words of the final captioning was subjected to training. In addition, the TF technique (mentioned in the related works section) was applied so that training can be accelerated by allowing the recurrent network to access the ground-truths during the inference process.

All the experiments presented in the next sections were obtained using a HPC node with an AMD EPYC 7742 64-Core Processor and a 40 Gb Nvidia A-100 graphic card.

A. HYPERPARMETER TUNING

As a first experiment, we maintain all the default hyperparameters of the model to study the impact of the different cost functions. Since the cross-entropy cost function was used to train the benchmark model, we contrasted the performance of the architecture using the negative log-likelihood (NLL), mean squared error (MSE), and the Kullback-Leibler Divergence (KLDIVLOSS) cost functions.

Once the first experimental phase is completed, the aim is to keep the cost function as an independent variable to sweep different optimizers. Once again, in addition to the optimizer used in the benchmark implementation (Adam), we examined the effect of AdamW, root mean square propogation optimizer (RMSprop), stochastic gradient descent (SGD), and Adadelta optimizers.

B. ENCODER ANALYSIS

In this scenario, once the optimal configuration of hyperparameters has been found, both the cost function and the network optimizer are set as fixed variables, allowing us to proceed with the second part of the experiment. Within this final stage, it is proposed to evaluate the performance of the architecture, both in terms of response quality and computational requirements, using different convolutional structures to replace the Oxford VGG model used in the encoder of the default implementation. The alternatives to be evaluated in this work correspond to the ResNet-101, ResNet-152, ResNeXt-101, and MobileNetV3 models.

Algorithm 1 Parameter Optimization and Training

Input: Set of images X, set of ground-truths Y, set of caption sizes S, initial learning rate γ, batch size β.
Output: Predicted caption $\boldsymbol{C}$, Set of individual attention masks $\boldsymbol{\alpha}$.
Initialization:
1: Initialize γ to 4e-4 and β to 32. ▷ Value of γ will depend on the training type.
2: Initial memory and hidden LSTM states are initialized by using separate MLPs given an image:
$\boldsymbol{c_0} = f_{init,c_0}(\frac{1}{L}\sum_{i=1}^{L} a_i)$
$\boldsymbol{h_0} = f_{init,h_0}(\frac{1}{L}\sum_{i=1}^{L} a_i)$
DATA ACQUISITION AND PRE-PROCESSING. (IN SECT. II-A.)
3: **Get** MSCOCO dataset ▷ From online server.
4: **for** each image **do**
5: Resize and Normalize.
6: **end for**
7: Sample a minibatch of m'_{tr} examples from the training set $\mathbb{B} = \{[\boldsymbol{X}^{(1)}: \boldsymbol{Y}^{(1)}], \ldots, [\boldsymbol{X}^{(m'_{tr})}: \boldsymbol{Y}^{(m'_{tr})}]\}$
CROSS-ENTROPY COST FUNCTION DEFINITION (SECT. V-A.)
8: $J(\boldsymbol{\theta}) = -\frac{1}{m'_{tr}}\sum_{i=1}^{m'_{tr}} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta})$
PARAMETER OPTIMIZATION FOR CONVOL. ENC.-DEC. (V-B.)
9: **while** stopping criterion not met **do**
10: Compute gradient estimate:
$\boldsymbol{g} \leftarrow \frac{1}{m'_{tr}}\nabla_{\boldsymbol{\theta}}\sum_{i=1}^{m'_{tr}} L\left(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}\right)$
11: Update parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{g}$
12: **end while**
CAPTION GENERATION OF UNSEEN IMAGE.
13: **Get** input image.
14: **Generate** the caption for the input image using optimized $\boldsymbol{\theta}$ parameters.
15: **Extract** caption matrix $\boldsymbol{C}$ and the set of masks $\boldsymbol{\alpha}$ from line 14.

C. TRANSFORMER-BASED APPROACHES

For this last experimental environment, the objective is to study the alternative of replacing the convolutional encoder of the original architecture by a model that dispenses with the traditional convolutional principles forged within the computer vision area, more specifically, focusing on incorporating transformer-based models in this specific part of the image captioning system. Despite its origin related to natural language processing, the ViT model demonstrated its viability for image classification tasks. Given the potential of this network to surpass our state-of-the-art, it was proposed as an experiment to verify the performance of such a model to carry out image captioning tasks. Consequently, it was decided to use both the original version of ViT and its version with distillation (DeiT).

It should be noted that since the present work does not require image classification tasks, both architectures were stripped of the last MLP layer since the attentional model will reuse the output of the transformer model. The schematization of the final model for image captioning is shown in Fig. 3.

Finally, it is worth mentioning that both the ViT and DeiT models correspond to models retrieved from the Huggingface repository, being pre-trained in the ImageNet-21k and ImageNet-1k datasets respectively.

On the one hand, the first method to be studied consists of defining $\gamma = 4e-4$ to train only the learnable parameters belonging to the decoder system architecture. This method is taken into consideration since the aim is to take advantage of the knowledge contained in the pre-trained models. By contrast, the second proposed methodology corresponds entirety with the previously described approach, with the difference that $\gamma = 1e-4$ is defined. Lastly, and as a final modality, we seek to rescue the model obtained with the second training experiment so that, in the last four iterations of the process, not only the decoder parameters are subjected to training, but also those that make up the last transformer block of both the ViT and DeiT models.

The final objective of this experiment was to use the BLEU-4 metric on both versions of the image captioning model to contrast the margin of improvement achieved concerning the state-of-the-art.

VII. RESULTS

From Table 5, it is possible to highlight an evident improvement in the performance of the model when using the cross-entropy as a loss function. Although the MSE loss is positioned as the second-best alternative throughout the experimental process, a difference of 31.584 in the Top-5 accuracy indicator and 0.187 in BLEU-4 metric shows a large gap between the cross-entropy function and this alternative. Considering this significant difference, the results obtained by the KLDIVLOSS and the NLL position them as unsuitable alternatives for the model to be trained on.

TABLE 5. Experimental results using *Top-5* accuracy and the *BLEU-4* performance metric for each one of the loss functions under study.

	Top-5 Accuracy	BLEU-4
Cross-entropy	**73.092**	**20.10**
MSE	41.508	1.40
KLDIVLOSS	32.186	1.173e-153
NLL	32.186	1.173e-153

TABLE 6. Experimental results using the training loss, the *Top-5* Accuracy, and the *BLEU-4* performance metrics for each one of the optimizers under study.

	Loss	Top-5 Accuracy	BLEU-4
Adam	**3.413**	**73.092**	**20.10**
AdamW	3.418	72.989	**20.10**
RMSprop	3.663	71.444	19.20
SGD	7.011	33.606	1.273e-153
Adadelta	7.133	33.045	1.272e-153

In addition to the quantitative results, Fig. 4 illustrates a captioning example generated using each one of the loss functions under study. The outcomes prove that the cross-entropy loss function is positioned not only as the one with the best results, but also the only loss function capable of generating a complete and meaningful description for an illustration that has never been seen by our model.

Proceeding with the second part of this scene, the results offered in Table 6 reveal a tighter situation when defining an optimal alternative. In the first instance, the optimizer Adam is positioned with the best results according to the three defined metrics. However, its variation, AdamW, not only returns the same BLEU-4 value as Adam, but it represents only a 0.005 and 0.133 of difference in the loss and Top-5 Accuracy indicators, respectively. This closeness in terms of results can be visualized using Fig. 5. In this illustration, each optimizer is tested by predicting the captioning for an image consisting of a child in front of a laptop computer. When contrasting both variations of the Adam optimizer, it is observed that the predictions only differ when mentioning the gender of the person in the image.

It is worth highlighting the performance of the RMSprop, which ranks as the third-best alternative, presenting a loss value of 3.663, along with 71.444 and 19.20 for Top-5 accuracy and BLEU-4, respectively. RMSprop shows promising results when comparing the output caption with the example image shown in Fig. 5. This optimizer is capable of generating a fully meaningful captioning by portrying to the content of the image. However, it missed minor details like not including a reference to the elderliness of the person in the illustration.

Finally, the SGD and Adadelta optimizers provided the worst results. Although both optimizers presented slightly different metrics, it is observed that neither of them were able to create a model capable of generating meaningful captions.

Now, referring to the results of the encoder testing phase shown in Table 7, two isolated analyses were conducted. At first, when looking for the convolutional model that

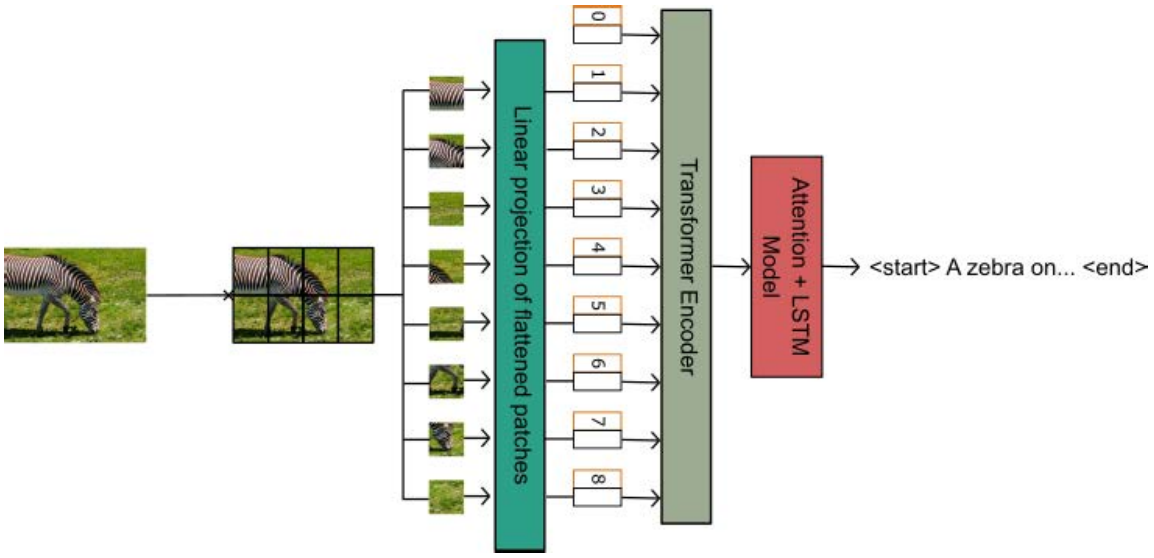
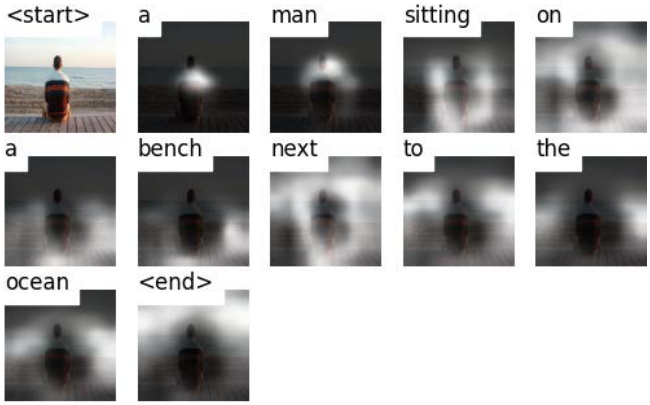


FIGURE 3. Overall representation of the ViT adaptation proposal for solving images captioning tasks. The MLP of the original implementation is replaced by the decoder used in previous experimental scenes.



(a) Image captioning result using cross entropy loss.

(b) Image captioning result using MSE loss.

(c) Image captioning result using NLL loss.

(d) Image captioning result using KLDIVLOSS.

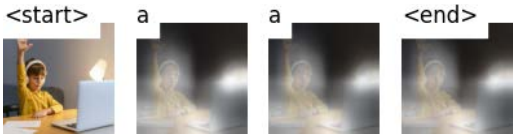
FIGURE 4. Image captioning results using an attention model with: (a) cross entropy loss, (b) MSE loss, (c) NLL loss, and (d) KLDIVLOSS. The results reveal an inadequate inference of MSE, NLL and KLDIVLOSS functions. By far, cross entropy is the only loss function that allows a proper training of our attention model.

allows the best captioning quality, the superiority of the ResNeXt-101 model is evidenced. This model stands out with a Top-5 Accuracy of 73.128 and a loss value of 3.404, surpassing the original encoder based on the VGG-16 architecture and the rest of the convolutional alternatives. On the other hand, the picture changes when looking for the architecture with lower computational requirements, trying to minimize the sacrifice of the output quality as much as possible. Therefore, MobileNetV3 demonstrates its inherent qualities as an architecture oriented to embedded environments, requiring 2,971,952 parameters, 3.5379 hours of training time, and 0.07975 seconds of average inference time. Such indicators become much more meaningful when referring to the BLEU-4, Top-5 Accuracy, and loss metrics, returning 19.50, 72.928, and 3.424, respectively.

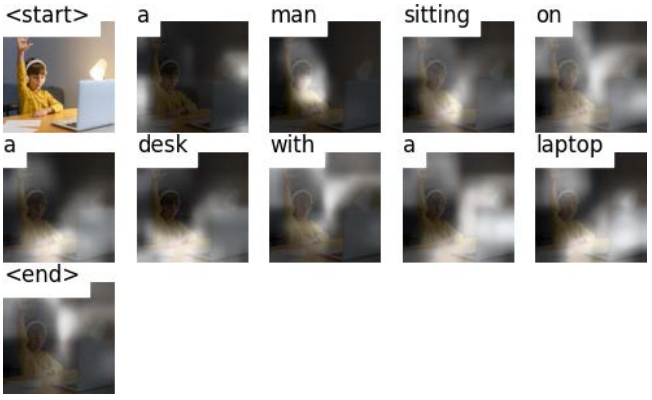
The evident closeness between the results, in terms of response quality, can be seen in the example of captioning included in Fig. 6. The ability of each of the models to

TABLE 7. Once the experimental phase has been completed with each proposed architecture for the system encoder, the quantitative results are shown. The chosen metrics denote both the quality of the response generated and the computational performance of each architecture.

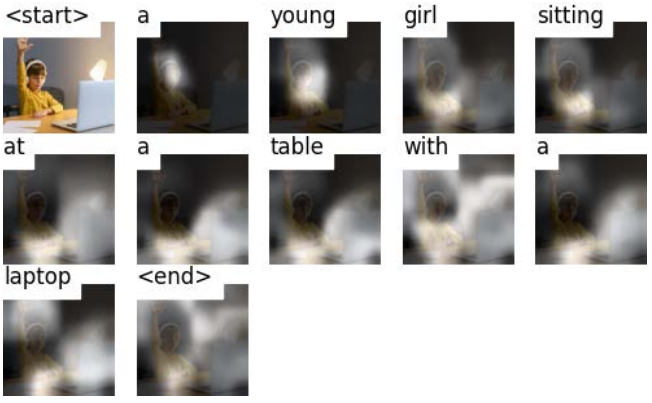
	BLEU-4	Top-5 Accuracy	Loss	Total Parameters	Training Time (Hours)	Inference Time (Seconds)	Computational Performance (GMAC's)
ResNet-101	20.10	73.092	3.413	42,500,160	5.6046	0.10765	7.85
ResNet-152	**20.20**	73.077	3.412	58,143,808	6.4077	0.14021	11.58
VGG-16	20.00	73.069	3.413	14,714,688	4.8353	0.08430	15.38
ResNeXt-101	19.80	**73.128**	**3.404**	86,742,336	7.8939	0.11023	16.5
MobileNet V3	19.50	72.928	3.424	**2,971,952**	**3.5379**	**0.07975**	**0.23**



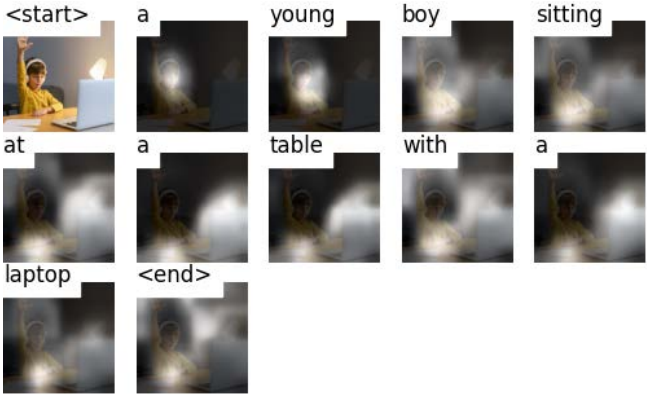
(a) Image captioning result using SGD and Adadelta optimizers.



(b) Image captioning result using RMSprop optimizer.



(c) Image captioning result using AdamW optimizer.



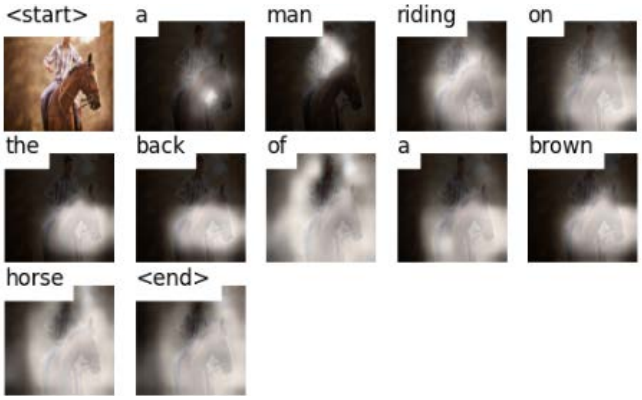
(d) Image captioning result using Adam optimizer.

FIGURE 5. Image captioning results using: (a) SGD and Adadelta optimizers, (b) RMSprop optimizer, (c) AdamW optimizer, and (d) Adam optimizer. The image illustrates the inadequate inference results of SGD and Adadelta when compared with their alternatives. Also, note that Adam optimizer yields the finest result over the test image (a recurrent outcome obtained for further experiments using images from the test set).

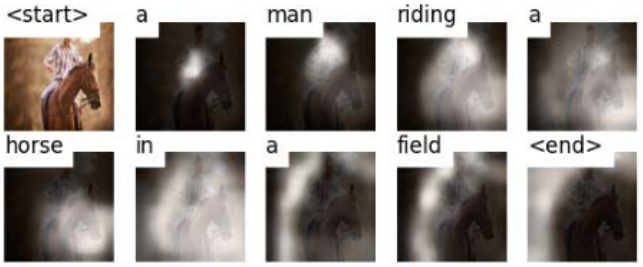
generate descriptions according to the scenario depicted in the input image, including different details regarding colors, positions, and environmental conditions, can be perceived. Likewise, this example provides a visualization of possible minor failures when generating the corresponding caption. In the aforementioned image, the encoder based on the VGG-16 architecture returns a description with redundancy, which can be justified by the training period established for the present experimentation.

Relying on a second example, Fig. 7 once again demonstrates the ability to generate a fully meaningful sentence by all architectures; however, not all of them manage to match the context of the image despite occasional errors in specific words. Under this scenario, the MobileNetV3 network generates an output that is entirely far from a possible ground-truth for the given image. Although this specific example is not a compelling reason to contradict the quantitative results previously shown, this example is intended to demonstrate a scenario where the robustness of a model for mobile environments becomes evident.

As for the results concerning the transformer-based architectures, Fig. 8 evidences the loss curves generated from the inference process on the validation group. Although both the ViT and DeiT based models show the lowest losses using the training method with the highest *gamma* value, it should be taken into account that from the fifth



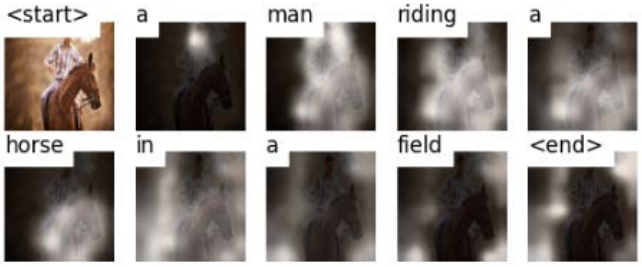
(a) Image captioning result using ResNet-152 as encoder.



(b) Image captioning result using ResNet-101 as encoder.

(c) Image captioning result using VGG-16 as encoder.

(d) Image captioning result using ResNext-101 as encoder.



(e) Image captioning result using MobileNet V.3 as encoder.

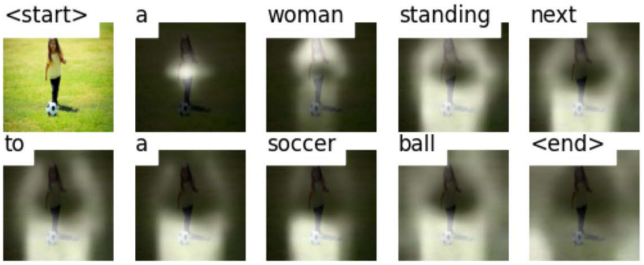
FIGURE 6. Image captioning results using as encoder: (a) ResNet-152, (b) ResNet-101, (c) VGG-16, (d) ResNext-101, and (e) MobileNet V.3. All the convolutional architectures allowed the generation of sentences with complete meaning matching considerably to the scenario presented in the input image. Reduced redundancy errors are appreciated when using VGG16.

iteration onwards, these models seem to suffer from possible overfitting. On the other hand, the loss curves behave more regularly throughout the iterations analyzed, showing little or no overfitting when using the alternative training methods. Therefore, beyond taking these values as indicators of the performance of the models, the aim is to show the evident convergence that exists throughout each training lapse.

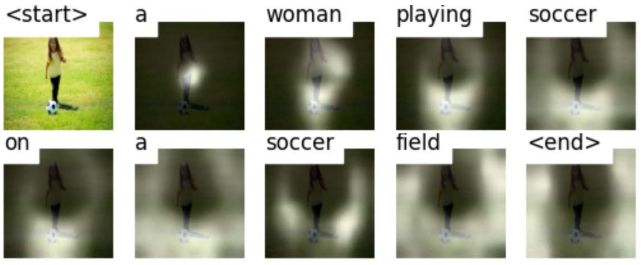
Having contemplated the convergence of the models, it is worthwhile to perform a similar visualization now using a metric related to the nature of natural language. Thus, Fig. 9 shows the evolution of the BLEU-4 with the passing of the iterations. Furthermore, within this graph, the results during the inference process on the validation set are shown. Therefore, when analyzing the impact of using a higher *gamma* value, both ViT and DeiT-based models present a relatively early learning *plateau* when reaching the fifth iteration. Conversely, the other two training methods present a significant improvement of BLEU-4. remaining in optimization even when reaching the last iterations. Both procedures allow a progressive improvement of the metric even during the last iterations; however, the methodology that contemplates the re-training of the transformer component stands out slightly.

However, considering that the inferences generated for the realization of this graph involved the use of TF, such values might not fully represent the capabilities of the models, since when seeking to caption an image devoid of a ground-truth, TF could not be applied. For this reason, it was decided to construct the results included in Fig. 10.

By employing much more realistic conditions for the inference process, it can be seen that the models trained with a lower γ outperformed the performance metrics of those

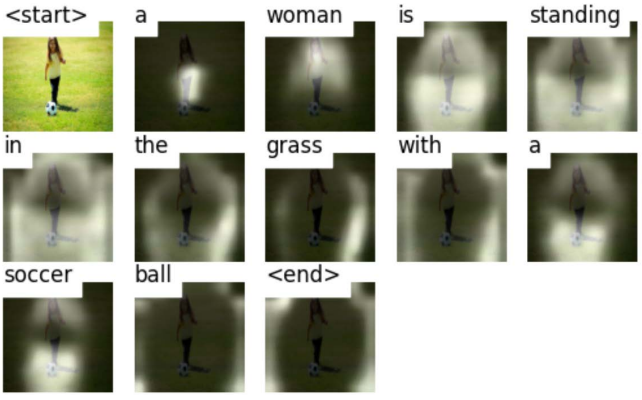


(a) Image captioning result using ResNet-152 as encoder.

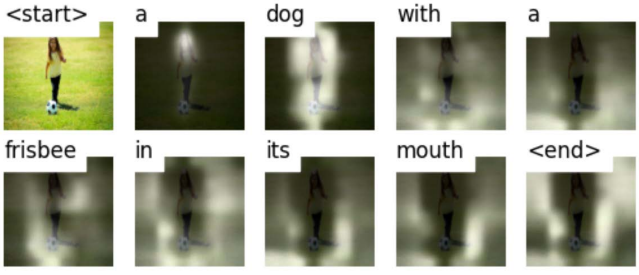


(b) Image captioning result using ResNet-101 as encoder.

(c) Image captioning result using VGG-16 as encoder.



(d) Image captioning result using ResNext-101 as encoder.



(e) Image captioning result using MobileNet V.3 as encoder.

FIGURE 7. Image captioning results using as encoder: (a) ResNet-152, (b) ResNet-101, (c) VGG-16, (d) ResNext-101, and (e) MobileNet V.3. It can be seen that the first four architectures generated results that were significantly close to the content of the input image. On the contrary, when using MobiliNet V. 3, the generated result consists of a description completely unrelated to the target scenario, even though the sentence was grammatically correct and made complete sense.

with a slightly higher γ in a very few number of iterations. Moreover, when using these results, a clear metrics boost is perceived, in contrast to when TF was used during inference. Thus, to contrast the best checkpoints obtained in each stage of this experimental scene, Table 8 allows to have a superior contrast of the maximum performance obtained when using ViT and DeiT through the application of each of the three training.

As a result, it can be verified that the use of TF during the inference process camouflaged the real performance of both models. Simulation results show that the DeiT-based model can be selected as the alternative with enhanced outcomes, specifically reaching a BLEU-4 of 34.44 through the training process involving the calculation of gradients for the last transformer block. Additionally, when reviewing the partial results of each training method, it is observed that regardless of the method applied, the DeiT-based model achieves the best BLEU-4 metrics.

TABLE 8. BLEU-4 metric obtained by the best checkpoint generated from each training process applied to the ViT and DeiT based models using a beam size of 3.

Encoder model (Inference modality)	Training 1	Training 2	Training 3
ViT (No TF)	32,07	32,98	33,24
ViT (TF)	23,07	23,09	23,24
DeiT (No TF)	**33,53**	**34,19**	**34,44**
DeiT (TF)	24,02	23,61	23,84

As a complement to the quantitative results shown above, Fig. 11 provides a brief sample of the accuracy that ViT- and DeiT-based models can provide when generating inference. The images used for this section were extracted from the

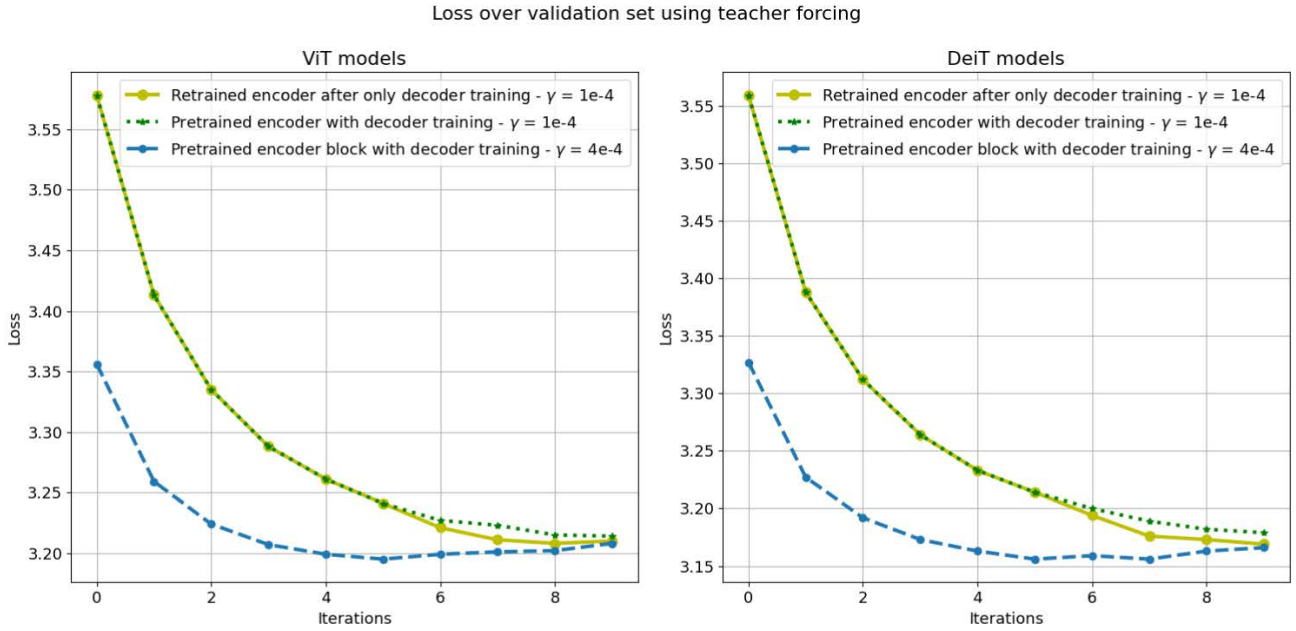


FIGURE 8. Evolution of the loss obtained during each of the corresponding iterations. These results were recovered using TF during the inference process on the validation set.

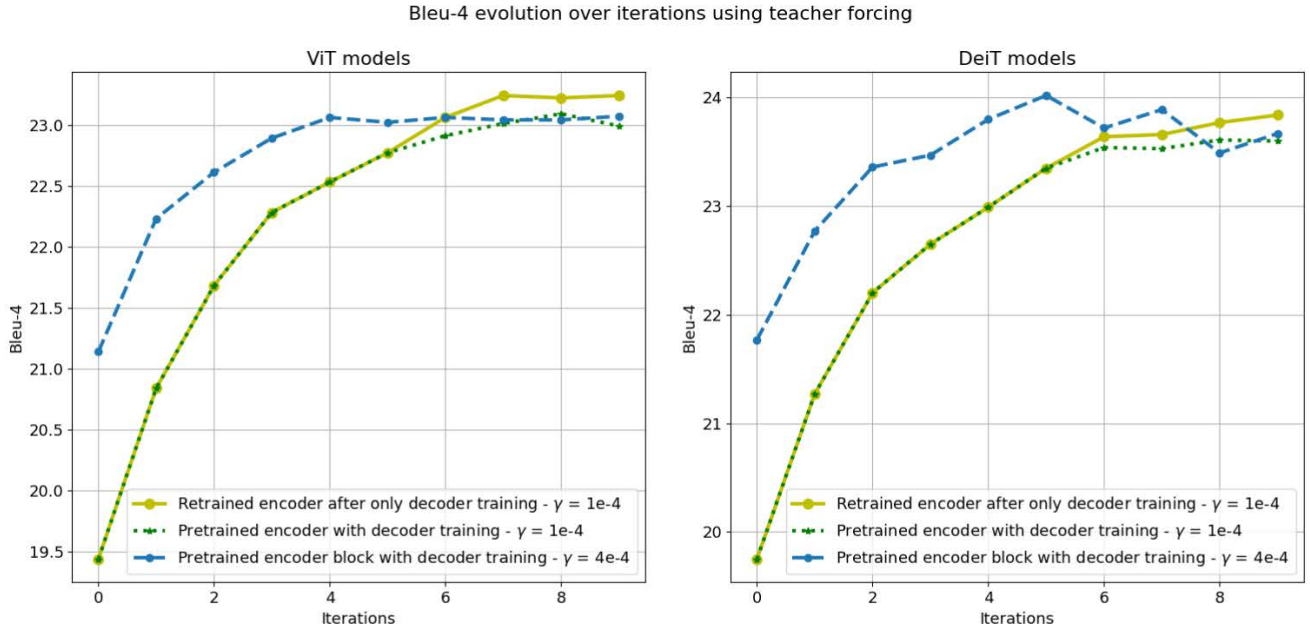


FIGURE 9. Evolution of the BLEU-4 metric obtained during each of the corresponding iterations. These results were recovered using TF during the inference process on the validation set.

validation set to use the ground truth linked to each image as a referential description.

Within this brief comparative scheme, we observe the ability of the models not only to describe relationships between objects or people, but also qualities related to the capture of physical aspects and generalization of similar entities. On the one hand, when working on the first image of Fig. 11, the DeiT model can not only denote the interaction of the dog with the frisbee, but it can also contribute with additional information about the colors of both entities. Also, when the image is presented with food, both models can recognize that the main content of the

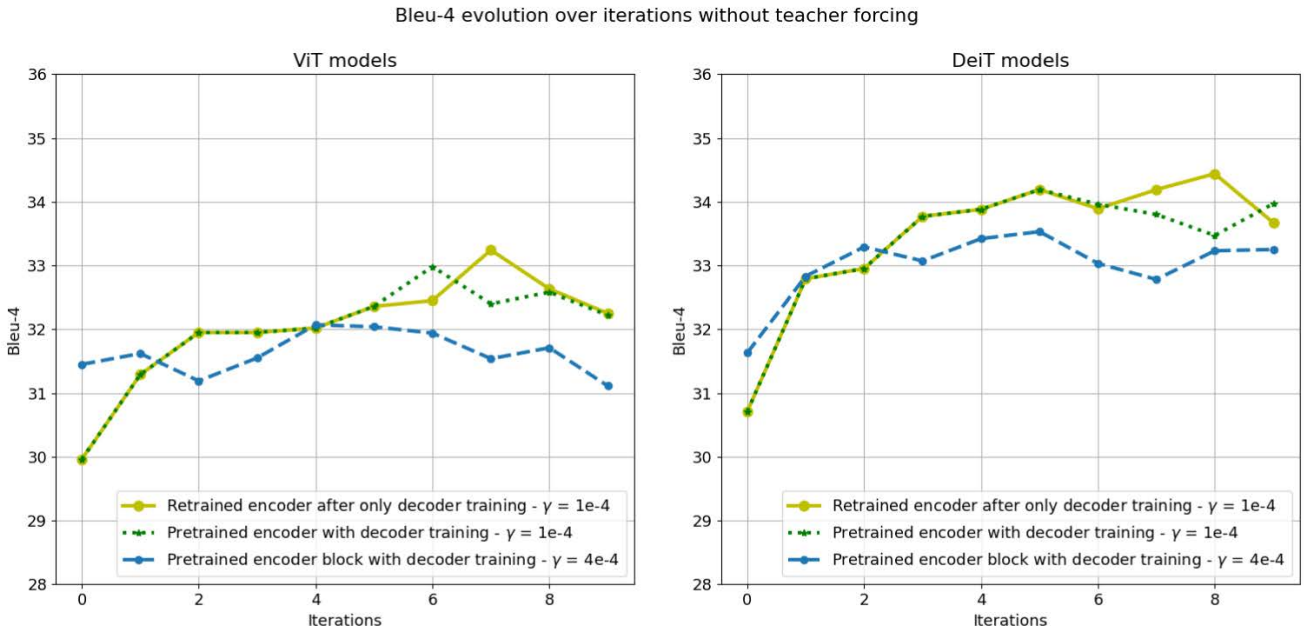


FIGURE 10. Evolution of the BLEU-4 metric obtained during each of the corresponding iterations. These results were retrieved without using TF during the inference process on the validation set.

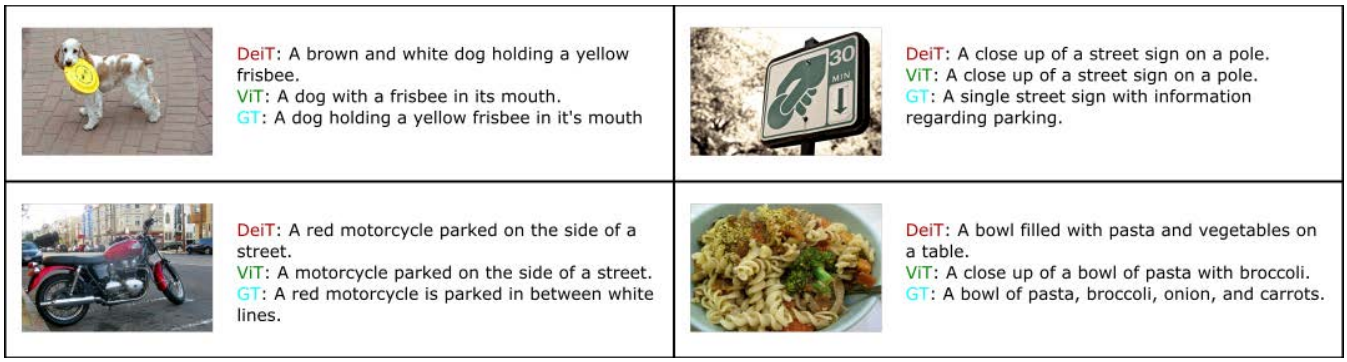


FIGURE 11. Examples of inference using images from the validation group. Models based on ViT and DeiT with best BLEU-4 metrics are used to contrast with the ground-truth provided by the dataset.

dish is pasta, however, the DeiT model can identify the presence of multiple vegetables within the dish, therefore, this architecture generalizes these foods into a single category.

VIII. CONCLUSION

During the first experimental stage, it was possible to determine that the cross-entropy was the loss function that achieved the best results, returning a Top-5 accuracy and BLEU-4 metrics of 73.092 and 0.201, respectively. On the other hand, once the loss function is set as an independent variable, the Adam optimizer returned the best indicators, completing the first training period with a loss value of 3.414, a Top-5 Accuracy of 73.092, and a BLEU-4 of 0.201. However, it is worth noting the good results obtained by the AdamW optimizer, matching in the BLEU-4 metric its Adam counterpart.

Furthermore, the comparative study focused on the convolutional model and its use as an encoder to yield two attractive alternatives depending on the final objective. On one hand, using the ResNeXt-101 architecture generated the best results in terms of response quality. This architecture returned values of 73.128 in Top-5 Accuracy, and 3.404 for the loss value, denoting an improvement with respect to the results obtained using VGG-16. On the other hand, when analyzing the models under lower computational demands, the encoder based on MobileNetV3 registered 2,971,952 parameters, a training time of 3.5379 hours, an inference time of 0.07975 seconds, and 0.23 GMACs. Thus, MobileNetV3 emerges as the most compact alternative without neglecting the quality of the generated captioning, which is evidenced by its great closeness in the BLEU-4, Top-5 Accuracy, and loss value metrics.

Regarding the study involving the use of transformer-based architectures as a replacement for convolutional models, both the ViT and DeiT models demonstrate their viability by verifying their convergence through the evolution of the loss throughout the iterations. In addition, the DeiT-LSTM model stands out as the alternative with the best BLEU-4 metric when trained in two phases: the first one in attempt to optimize only the decoder parameters, and the second phase incorporating the parameters of the last transformer block to be optimized using a value of $\gamma = 1e-4$. As a result, the model achieved a BLEU-4 of 34.44, surpassing the state-of-the-art from the paper *Show, Attend and Tell*, whose best results consisted in a BLEU-4 of 24.3 in its *soft-attention* based model, and 25.0 for its *hard-attention* alternative.

Although we have proved that the three optimizers and two encoder options offer feasible results for this architecture, future works can benefit from the individual training epoch to further study the convergence pace of the model under limited edge-computational devices. In addition, future researchers can study the viability of not only using different encoder architectures than the presented ones, but also analyze the impact of other alternatives to LSTM models for the decoding step, together with an extended investigation on the architectural frameworks. Another element concerning the training stage of our model is the decision to use the MSCOCO 2014 dataset. The selection was made based on: i) the need of a large image set, and ii) the need to replicate the results of the benchmark paper. However, both the convolutional and transformer-based variants have potential for further research, where the reader can study the performance and behavior of our model when trained with other datasets such as Flickr8k or Flickr30k. Finally, another alternative to foster this work would be to include further hyperparameters to the study (e.g., dropout rate, batch size, different types of stride and pooling, size of the kernels, weight initialization methods, model depth, weight decay, etc.). Also, different methodologies of optimization such as Random Search, Grid Search, etc can be applied supported by a Hyperparameter Tuning Framework, enabling an in-depth research of the attention architecture

REFERENCES

[1] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7008–7024.

[2] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Babytalk: Understanding and generating simple image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2891–2903, Oct. 2013.

[3] Y. Yang, C. Teo, H. Daumé, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 444–454.

[4] K. Kpalma and J. Ronsin, "An overview of advances of pattern recognition systems in computer vision," *Vis. Syst.*, vol. 4, p. 26, May 2007.

[5] C. G. Amza and D. T. Cicic, "Industrial image processing using fuzzy-logic," *Proc. Eng.*, vol. 100, pp. 492–498, Oct. 2015.

[6] A. Rastogi, R. Arora, and S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic," in *Proc. 2nd Int. Conf. signal Process. Integr. Netw. (SPIN)*, 2015, pp. 500–505.

[7] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.

[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2048–2057.

[9] D. Carrión-Ojeda, R. Fonseca-Delgado, and I. Pineda, "Analysis of factors that influence the performance of biometric systems based on eeg signals," *Expert Syst. Appl.*, vol. 165, Feb. 2021, Art. no. 113967. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742030748X

[10] R. Castro, I. Pineda, and M. E. Morocho-Cayamcela, "Hyperparameter tuning over an attention model for image captioning," in *Information Communication Technology*, J. P. Salgado Guerrero, J. C. Espinosa, M. C. Lozada, and S. Berrezueta-Guzman, Eds. Cham, Switzerland: Springer, 2021, pp. 172–183.

[11] H. Larochelle and G. Hinton, "Learning to combine foveal glimpses with a third-order Boltzmann machine," in *Advance Neural Information Processing System*, vol. 1, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2010, pp. 1243–1251.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Tech. Rep., 2016.

[13] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.

[14] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," 2015, *arXiv:1412.7755*.

[15] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 595–603.

[16] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (M-RNN)," 2015, *arXiv:1412.6632*.

[17] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[18] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advance Neural Information Processing System*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[20] C. Wang, S. Wu, and S. Liu, "Accelerating transformer decoding via a hybrid of self-attention and recurrent neural network," 2019, *arXiv:1909.02279*.

[21] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," Tech. Rep., 2018.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019, *arXiv:1810.04805*.

[23] J. W. Chen, X. K. Sigalingging, J.-S. Leu, and J.-I. Takada, "Applying a hybrid sequential model to Chinese sentence correction," *Symmetry*, vol. 12, no. 12, p. 1939, 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/12/1939

[24] A. Patel and A. Varier, "Hyperparameter analysis for image captioning," 2020, *arXiv:2006.10923*.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," Tech. Rep., 2021.

[26] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "CPTR: Full transformer network for image captioning," 2021, *arXiv:2101.10804*.

[27] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 13041–13049.

[28] H. Lu, R. Yang, Z. Deng, Y. Zhang, G. Gao, and R. Lan, "Chinese image captioning via fuzzy attention-based DenseNet-BiLSTM," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 1, Mar. 2021, Art. no. 48, doi: 10.1145/3422668.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755.
[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2009, pp. 248–255.
[33] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning to improve multi-hop searching and extended wireless reachability in V2X," *IEEE Commun. Lett.*, vol. 24, no. 7, pp. 1477–1481, Sep. 2020.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

ROBERTO CASTRO (Student Member, IEEE) is currently pursuing the degree in IT with Yachay Tech University, Urcuquí, Ecuador. Since 2021, he has been affiliated with the DeepArc Research Group, in which he has started his career as a Young Researcher, making his first publication in *Communications in Computer and Information Science*, and he collaborates as a Volunteer Researcher with the SDAS Research Group.

His research interests include deep learning, computer vision, natural language processing, and data science.

ISRAEL PINEDA (Member, IEEE) received the B.E. degree in computer systems from Universidad Politécnica Salesiana, Ecuador, in 2011, the M.S. degree in computer science and engineering from Chonbuk National University, South Korea, in 2015, and the Ph.D. degree, in 2018. He is currently a full-time Professor with Yachay Tech University, Ecuador. His research interests include computer graphics, fluid simulations, physically-based simulation, rendering, and scientific computing.

WANSU LIM (Member, IEEE) received the M.Sc. and Ph.D. degrees from the Gwangju Institute Science and Technology (GIST), South Korea, in 2007 and 2010, respectively. He is currently an Associate Professor with the Kumoh National Institute of Technology (KIT), South Korea, leading the activities of the communication technologies, machine learning research, and computer vision processing. From 2010 to 2013, he was a Research Fellow with the University of Hertfordshire, U.K., and then a Postdoctoral Researcher with the Institut national de la recherche scientifique (INRS), Canada, from 2013 to 2014. He is also a Technical Committee Member of Elsevier Computer Communications and a member of IEEE Communications Society. He has authored over 60 peer-reviewed journals and conference papers and served as a reviewer for several IEEE conferences and journals.

MANUEL EUGENIO MOROCHO-CAYAMCELA (Member, IEEE) received the B.S. degree in electronic engineering from Universidad Politécnica Salesiana, Cuenca, Ecuador, in 2012, the M.Sc. degree in communications engineering and networks from The University of Birmingham, England, U.K., in 2016, and the Ph.D. degree in electronic engineering from the Kumoh National Institute of Technology, Gumi-si, Republic of Korea.

From 2017 to 2020, he was a Senior Researcher with the KIT Future Communications and Systems Laboratory, Gumi-si. Since 2020, he has been working as a Fellow Researcher with ESPOL, Guayaquil, Ecuador. He is currently a full-time Professor with Yachay Tech University, Urcuquí, Ecuador. His research interests include artificial intelligence, deep learning, computer vision, wireless communications, and optimization.

Mr. Morocho-Cayamcela was a recipient of the SENESCYT Fellowship from The National Secretariat for Higher Education, Science, Technology and Innovation of Ecuador, in 2015, and the KIT Doctoral Grant from the Kumoh National Institute of Technology, in 2017.