Hyperparameter Tuning over an Attention Model for Image Captioning

Chapter · November 2021

DOI: 10.1007/978-3-030-89941-7_13

3 authors:

Roberto Castro
Yachay Tech
2 PUBLICATIONS 0 CITATIONS
SEE PROFILE

Israel Pineda
Yachay Tech University
27 PUBLICATIONS 25 CITATIONS
SEE PROFILE

Manuel Eugenio Morocho-Cayamcela
Yachay Tech
45 PUBLICATIONS 300 CITATIONS
SEE PROFILE

Hyperparameter Tuning over an Attention Model for Image Captioning

Roberto Castro, Israel Pineda,
and Manuel Eugenio Morocho-Cayamcela(✉)

School of Mathematical and Computational Sciences,
Deep Learning for Autonomous Driving, Robotics, and Computer Vision Research
Group (DeepARC), Yachay Scientific Computing Group (SCG),
Yachay Tech University, Hda. San José s/n y Proyecto Yachay,
100119 Urcuquí, Ecuador
{roberto.castro,ipineda,mmorocho}@yachaytech.edu.ec
http://www.yachaytech.edu.ec

Abstract. Considering the historical trajectory and evolution of image captioning as a research area, this paper focuses on *visual attention* as an approach to solve captioning tasks with computer vision. This article studies the efficiency of different hyperparameter configurations on a state-of-the-art visual attention architecture composed of a pre-trained residual neural network encoder, and a long short-term memory decoder. Results show that the selection of both the cost function and the gradient-based optimizer have a significant impact on the captioning results. Our system considers the cross-entropy, Kullback-Leibler divergence, mean squared error, and the negative log-likelihood loss functions, as well as the adaptive momentum, AdamW, RMSprop, stochastic gradient descent, and Adadelta optimizers. Based on the performance metrics, a combination of cross-entropy with Adam is identified as the best alternative returning a Top-5 accuracy value of 73.092, and a BLEU-4 value of 0.201. Setting the cross-entropy as an independent variable, the first two optimization alternatives prove the best performance with a BLEU-4 metric value of 0.201. In terms of the inference loss, Adam outperforms AdamW with 3.413 over 3.418 and a Top-5 accuracy of 73.092 over 72.989.

Keywords: Image captioning · Visual attention · Computer vision · Supervised learning · Artificial intelligence

1 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 [17]. 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 [10,20].

J. P. Salgado Guerrero et al. (Eds.): TICEC 2021, CCIS 1456, pp. 172–183, 2021.
https://doi.org/10.1007/978-3-030-89941-7_13

As a consequence, 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 [21]. Several researchers have replicated the benchmark implementation proposed by Xu et al. for further study [19]. 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. On the other hand, 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 [3].

In this paper, we study the performance impact of different hyperparameters of the model during the training and testing stages. More specifically, we conduct a comparative study to select the cost function that minimizes the training error over a certain number of epochs for our specific application. In addition, and using the leading cost function as an independent variable, we execute an optimizer sweep to appoint the best possible hyperparameter configuration for our captioning task. Based on our results, we can confidently claim that the arrangement of the cross-entropy as a cost function along with the gradient-based Adam optimizer, have led to superior results in terms of the top-5 accuracy and BLEU-4 metrics.

2 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 [2]. 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. [14], and the one proposed by Ba et al. [1].

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)

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 [9]. 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. [13], Vinyals et al. [18], and Donahue et al. [5] have reused this idea in their respective research efforts.

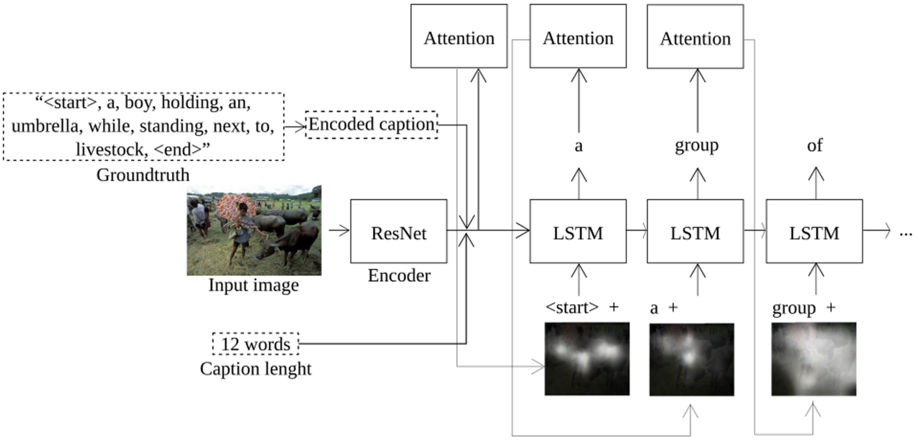


Fig. 1. Overall representation of the convolutional encoder-decoder architecture built to generate real captioning. The model uses a pre-trained ResNet architecture as the encoder backbone, along with recurrent LSTM operations for the decoder. The objects with discontinuous contours are only used during the training stage.

3 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.

On the one hand, the encoder makes use of transfer learning by borrowing the convolutional architecture of Resnet [6]. This incorporation aims to generate an encoded version of the input RGB image, with an output dimensionality of $2048 \times 14 \times 14$. On the decoder side, given the sequential nature of the problem to be solved, an LSTM recursive architecture is constructed [7]. Up to this point, the description of the input image is generated in a word-by-word basis. At each decoding step, the attention network uses the output generated by the encoder together with the previous hidden state, generating averaged weights for each pixel of the encoded image. Therein, the network is told "where" to look by assigning higher weights to pixels of upward relevance. Using this outcome and the previously generated word as references, the LSTM network generates a definitive caption for the input image.

On the other hand, the objects in Fig. 1 denoted with discontinuous contours are 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.

3.1 The Dataset Structure

The dataset used for training the network was the 2014 version of the MS COCO variant oriented to image captioning tasks [12]. 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.

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 [4], 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 $\boldsymbol{X}\{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(m)}\}$, where each image $\boldsymbol{X}^{(i)}$ is mapped to a ground truth caption $\boldsymbol{Y}^{(i)}$ that represents the corresponding ground-truth encoded caption.

Encoded Captions. In order to be able to manipulate the descriptions associated with each image in the dataset, the model uses a `.json` mapping file. Within this file, each word used in the captioning of the entire dataset has an identification number. Thus, the complete vocabulary supported by the network and its numerical equivalents can be visualized in this file. This new `.json` file will contain an array where each of its elements will correspond to the word-by-word captioning of each image using the numerical equivalences defined within the mapping file.

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.

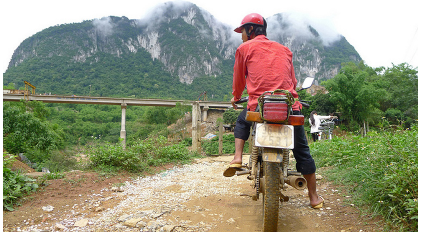


Fig. 2. Image taken from the training set with an associated groundtruth caption: "`a man with a red helmet on a small moped on a dirt road`".

Table 3. Mapping system used to encode the caption the example image.

Original 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

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:

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

considering that it has been generated from the equivalences contained in the mapping file, the contents of which are presented in Table 3.

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.

3.2 Hyperparameter Tuning

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 $J(\boldsymbol{\theta})$ [15].

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:

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

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 $\mathbf{C}$.

Adaptive Moment Optimizer. In order to optimize our attention model through a gradient-based optimization method, we express the gradient vector of (1) with respect to θ as

$$\begin{aligned} \mathbf{g} &= \nabla_{\theta} J(\boldsymbol{\theta}) \\ &= \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= \frac{1}{m}\sum_{i=1}^{m} \Big(\hat{p}^{(i)} - \boldsymbol{Y}^{(i)}\Big)\boldsymbol{X}^{(i)}. \end{aligned} \tag{2}$$

To locate the minimum of $J(\boldsymbol{\theta})$, the proposed optimization algorithm moves to the negative direction of (2) iteratively. Our model computes individual adaptive learning rates for different parameters from estimates of first and second moments of $\mathbf{g}$ [8].

4 Experimental Settings

This work proposes two experimental scenarios. First, 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, mean squared error, and the Kullback-Leibler Divergence 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, RMSprop, SGD, and Adadelta optimizers.

Both experimental phases were applied over one training epoch, using a workstation with 8 GB of RAM and an NVIDIA GTX1650 graphical processing unit (GPU).

The criterion used to contrast the performance of the different hyperparameter settings consisted in the interpretation of the top-5 accuracy, loss value, and BLEU-4 as a way to compute the similarity between the predicted captions and the available ground-truths (based on 4-grams modified comparisons) [16]. In addition, and in a non-quantitative way, we have considered the quality of the generated captions for specific unseen images.

5 Results

From Table 4, 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 mean squared error (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 Kullback–Leibler divergence (KLDIVLOSS) and the negative log-likelihood loss (NLL), position them as unsuitable alternatives for the model to be trained on.

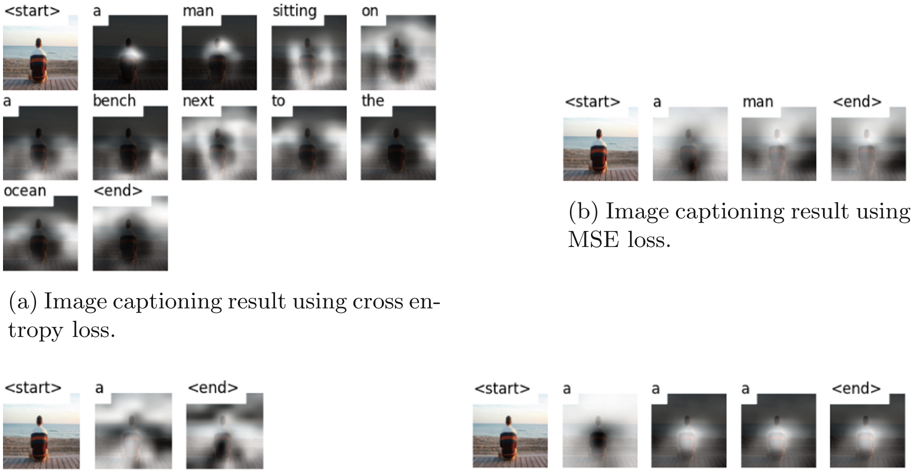
Table 4. Experimental results using *Top-5* accuracy and the *BLEU-4* performance metric for each one of the loss functions under study.

	Top-5 Accuracy†	BLEU-4†
Cross-entropy	**73.092**	**0.201**
MSE	41.508	0.014
KLDIVLOSS	32.186	1.173e-155
NLL	32.186	1.173e-155

†Trained using a workstation with 8 GB of RAM and an NVIDIA GTX1650 GPU.

In addition to the quantitative results, Fig. 3 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 experimental scene, the results offered in Table 5 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



(a) Image captioning result using cross entropy loss.

(b) Image captioning result using MSE loss.

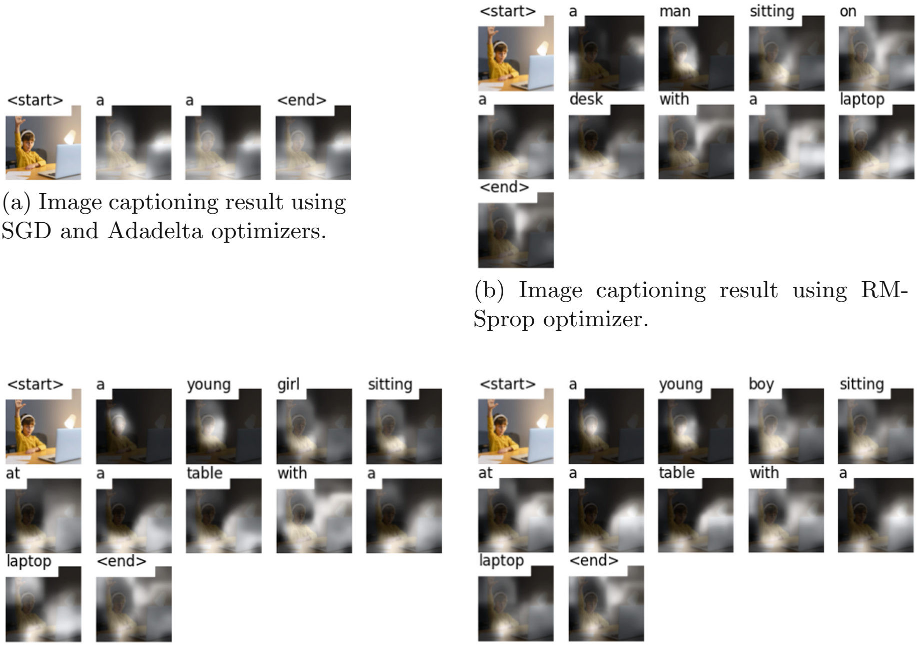
(c) Image captioning result using NLL loss.

(d) Image captioning result using KL-DIVLOSS.

Fig. 3. 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.

same BLEU-4 value as Adam, but it is only 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. 4. 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 root mean square propogation optimizer (RMSprop), which ranks as the third-best alternative, presenting a loss value of 3.663, along with 71.444 and 0.192 for Top-5 accuracy and BLEU-4, respectively. RMSprop shows promising results when comparing the output caption with the example image shown in Fig. 4. 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 stochastic gradient descent (SGD) and the 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.



(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.

Fig. 4. 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).

Table 5. 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**	**0.201**
AdamW	3.418	72.989	**0.201**
RMSprop	3.663	71.444	0.192
SGD	7.011	33.606	1.273e-155
Adadelta	7.133	33.045	1.272e-155

[†] Trained using a workstation with 8 GB of RAM and an NVIDIA GTX1650 GPU.

6 Conclusions and Discussion

During the experimental stage, it was possible to determine that the cross-entropy was the loss function with the best results, returning 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, the results obtained are tight close to the outcomes obtained with the AdamW optimizer, sharing the same BLEU-4 value. The training time required for each epoch was six hours, i.e., a total of 48 h was required for the generation of all the results.

Although we have proved that the three optimizers 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 computational resources. In addition, future works can study the viability of using a different encoder architecture than ResNet, together with an extended investigation on the architectural framework. 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.), enabling an in-depth research of the attention architecture.

Acknowledgments. This research was partially supported by the Ecuadorian Corporation for the Development of Research and Academia (CEDIA), under the project "Divulga Ciencia 2021", grant No.: JLE-CN-2021-0105. This research was partially supported by NVIDIA Corporation under the program "NVIDIA Jetson Nano 2GB Developer Kit".

References

1. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. CoRR abs/1412.7755 (2014). http://dblp.uni-trier.de/db/journals/corr/corr1412.html#BaMK14
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, 07–09 May 2015 (2015)
3. Carrión-Ojeda, D., Fonseca-Delgado, R., Pineda, I.: Analysis of factors that influence the performance of biometric systems based on EEG signals. Expert Syst. Appl. **165**, 113967 (2021) https://doi.org/10.1016/j.eswa.2020.113967. https://www.sciencedirect.com/science/article/pii/S095741742030748X
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
5. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)

6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference for Learning Representations (2015)
9. Kiros, R., Salakhutdinov, R., Zemel, R.: Multimodal neural language models. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 595–603. PMLR, Bejing (2014). https://proceedings.mlr.press/v32/kiros14.html
10. Kulkarni, G., et al.: Baby talk: understanding and generating simple image descriptions. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2891–2903 (2013). https://doi.org/10.1109/TPAMI.2012.162
11. Larochelle, H., Hinton, G.: Learning to combine foveal glimpses with a third-order Boltzmann machine. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems, vol. 1, pp. 1243–1251. Curran Associates, Inc. (2010)
12. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
13. Mao, J., Xu, W., Yang, Y., Wang, J., Yuille, A.L.: Deep captioning with multimodal recurrent neural networks (m-RNN). In: International Conference for Learning Representations (2015). http://arxiv.org/abs/1412.6632
14. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, pp. 2204–2212 (2014)
15. Morocho-Cayamcela, M.E., Lee, H., Lim, W.: Machine learning to improve multi-hop searching and extended wireless reachability in V2X. IEEE Commun. Lett. **24**(7), 1477–1481 (2020). https://doi.org/10.1109/LCOMM.2020.2982887
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL 2002 (2001). https://doi.org/10.3115/1073083.1073135
17. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
18. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
19. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention (2016)
20. Yang, Y., Teo, C., Daumé III, H., Aloimonos, Y.: Corpus-guided sentence generation of natural images. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 444–454 (2011)
21. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)