UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: Image captioning based on a visual attention approach

Trabajo de integración curricular presentado como requisito para la obtención del título de ingeniero en tecnologías de la información

Autor/a:

Castro Izurieta Roberto Raúl

Tutor/a:

Ph.D. - Morocho Cayamcela Manuel Eugenio

Urcuquí, Julio y 2022

Urcuquí, 17 de junio de 2022

SECRETARÍA GENERAL
(Vicerrectorado Académico/Cancillería)
ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN
ACTA DE DEFENSA No. UITEY-ITE-2022-00012-AD

A los 17 días del mes de junio de 2022, a las 11:30 horas, de manera virtual mediante videoconferencia, y ante el Tribunal Calificador, integrado por los docentes:

Presidente Tribunal de Defensa	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.
Miembro No Tutor	Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.
Tutor	Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.

El(la) señor(ita) estudiante **CASTRO IZURIETA, ROBERTO RAUL**, con cédula de identidad No. **0922799481**, de la **ESCUELA DE CIENCIAS MATEMÁTICAS Y COMPUTACIONALES**, de la Carrera de **TECNOLOGÍAS DE LA INFORMACIÓN**, aprobada por el Consejo de Educación Superior (CES), mediante Resolución **RPC-SO-43-No.496-2014**, realiza a través de videoconferencia, la sustentación de su trabajo de titulación denominado: **Image captioning based on a visual attention approach** , previa a la obtención del título de **INGENIERO/A EN TECNOLOGÍAS DE LA INFORMACIÓN**.

El citado trabajo de titulación, fue debidamente aprobado por el(los) docente(s):

Tutor	Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.

Y recibió las observaciones de los otros miembros del Tribunal Calificador, las mismas que han sido incorporadas por el(la) estudiante.

Previamente cumplidos los requisitos legales y reglamentarios, el trabajo de titulación fue sustentado por el(la) estudiante y examinado por los miembros del Tribunal Calificador. Escuchada la sustentación del trabajo de titulación a través de videoconferencia, que integró la exposición de el(la) estudiante sobre el contenido de la misma y las preguntas formuladas por los miembros del Tribunal, se califica la sustentación del trabajo de titulación con las siguientes calificaciones:

Tipo	Docente	Calificación
Presidente Tribunal De Defensa	Dr. IZA PAREDES, CRISTHIAN RENE , Ph.D.	10,0
Tutor	Dr. MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.	10,0
Miembro Tribunal De Defensa	Dr. ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.	10,0

Lo que da un promedio de: **10 (Diez punto Cero)**, sobre 10 (diez), equivalente a: **APROBADO**

Para constancia de lo actuado, firman los miembros del Tribunal Calificador, el/la estudiante y el/la secretario ad-hoc.

Firmado electrónicamente por:
ROBERTO RAUL CASTRO IZURIETA

CASTRO IZURIETA, ROBERTO RAUL
Estudiante

Firmado electrónicamente por:
CRISTHIAN RENE IZA PAREDES

IZA PAREDES, CRISTHIAN RENE , Ph.D.
Presidente Tribunal de Defensa

Firmado electrónicamente por:
MANUEL EUGENIO MOROCHO CAYAMCELA

MOROCHO CAYAMCELA, MANUEL EUGENIO , Ph.D.
Tutor

Firmado electrónicamente por:
JULIO JOAQUIN ARMAS ARCINIEGA

ARMAS ARCINIEGA, JULIO JOAQUIN , Ph.D.
Miembro No Tutor

DAYSY MARGARITA MEDINA BRITO

Firmado digitalmente por DAYSY MARGARITA MEDINA BRITO
DN: CN=DAYSY MARGARITA MEDINA BRITO, SERIALNUMBER=151220162727, OU=ENTIDAD DE CERTIFICACION DE INFORMACION, O=SECURITY DATA S.A. 2, C=EC
Razón: Soy el autor de este documento
Ubicación: la ubicación de su firma aquí
Fecha: 2022.06.20 12:57:10-05'00'
Foxit PDF Editor Versión: 11.2.1

MEDINA BRITO, DAYSY MARGARITA
Secretario Ad-hoc

Autoría

Yo, **Roberto Raúl Castro Izurieta**, con cédula de identidad 0922799481, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, mes y año.

Firmado electrónicamente por:
ROBERTO RAUL CASTRO IZURIETA

Roberto Raúl Castro Izurieta

CI: 0922799481

Autorización de publicación

Yo, **Roberto Raúl Castro Izurieta**, con cédula de identidad 0922799481, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, Julio y 2022.

Firmado electrónicamente por:
ROBERTO RAUL
CASTRO
IZURIETA

Roberto Raúl Castro Izurieta

CI: 0922799481

Dedication

Dedicated to my family, to the teachers who trusted me, and finally to the people who denied the blessing that a university like Yachay Tech represents for Ecuador and the scientific community. We are Yachay Tech.

Roberto Raúl Castro Izurieta

Acknowledgment

I thank my parents for their infinite love, their tireless support, their invaluable parenting and their endless desire to see their children living what they love and always nourishing themselves with the valuable knowledge that this world holds.

I thank my tutor, Manuel Eugenio Morocho, for his tremendous support for my development as a young researcher.

Finally, I thank SDAS Research Group, together with Dr. Wansu Lim of Kumoh National Institute of Technology, for trusting and seeing in me a student and researcher of great value.

Roberto Raúl Castro Izurieta

Resumen

Este trabajo se centra en *atención visual*, un enfoque de vanguardia para las tareas de subtitulación de imágenes dentro del área de visión por ordenador. Estudiamos el impacto, en términos de eficiencia, que generan diferentes configuraciones de hiperparámetros en una arquitectura de atención visual codificadora-decodificadora. Los resultados muestran que la correcta selección tanto de la función de coste como del optimizador basado en el gradiente puede tener un impacto significativo en los resultados del subtitulado. Nuestro sistema considera las funciones de pérdida de entropía cruzada, divergencia de Kullback-Leibler, error medio al cuadrado y log-verosimilitud negativa, así como los optimizadores de momento adaptativo (Adam), AdamW, RMSprop, descenso de gradiente estocástico y Adadelta. Tras la experimentación, se identifica una combinación de entropía cruzada con Adam como la mejor alternativa que devuelve un valor de precisión Top-5 de 73,092, y un BLEU-4 de 20,10. Además, se realizó un análisis comparativo de arquitecturas convolucionales alternativas para demostrar su rendimiento como codificador. Nuestros resultados muestran que ResNext-101 destaca con una precisión Top-5 de 73,128, y un BLEU-4 de 19,80; posicionándose como la mejor opción cuando se busca la calidad óptima de subtitulado. Sin embargo, MobileNetV3 demostró ser una alternativa mucho más compacta con 2.971.952 parámetros y 0,23 giga de operaciones de multiplicación-acumulación de punto fijo por segundo (GMAC). En consecuencia, MobileNetV3 ofrece una calidad de salida competitiva a costa de un menor rendimiento computacional, respaldado por los valores de 19,50 y 72,928 para el BLEU-4 y el Top-5 Accuracy, respectivamente. Por último, al probar los modelos transformador de visión (ViT), y transformador de imagen con eficiencia de datos (DeiT) para sustituir el componente convolucional de la arquitectura, DeiT logró una mejora sobre ViT, obteniendo un valor de 34,44 en la métrica BLEU-4.

Palabras Clave:

Subtitulación de imágenes, atención visual, visión por computadora, aprendizaje supervisado, inteligencia artificial.

Abstract

This thesis focuses on *visual attention*, a state-of-the-art approach for image captioning tasks within the computer vision research area. We study the impact, in terms of efficiency, that different hyperparemeter configurations generate on an encoder-decoder visual attention architecture. Results show that the correct selection of both the cost function and the gradient-based optimizer can have a significant impact in the captioning results. Our system considers the cross-entropy, Kullback-Leibler divergence, mean squared error, and negative log-likelihood loss functions, as well as the adaptive momentum (Adam), AdamW, RMSprop, stochastic gradient descent, and Adadelta optimizers. After experimentation, 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 of 20.10. Further, a comparative analysis of alternative convolutional architectures was conducted to demonstrate 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.

Keywords:
Image captioning, visual attention, computer vision, supervised learning, artificial intelli-

gence.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

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, 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].

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. First, 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 work, 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 experi-

mentation. 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.

1.2 Problem statement

The problem lies in the historical and current landscape of the image captioning area. Research branches such as image classification, present results that are close to the maximum possible values for accuracy metrics for the most famous challenges such as MNIST or Imagenet [11, 12]. In contrast, the most reputed approaches within the image captioning world are far from being close to a hypothetical perfection in terms of metrics [13, 14, 8].

Among the aspects that reinforce the demand for solving this problem, is the role played by image captioning in image indexing. It represents a large part of the contribution made by the development of technologies in this area. The fact that a machine is capable of understanding high-level details from basic features, not only makes it possible to increase the automation of multimedia content in social networks, but also to boost sectors such as education, commerce or biomedicine [15, 16].

1.3 Objectives

1.3.1 General Objective

The general objective is to improve the performance records established by the *Show, Attend and Tell* architecture [8], either through conservative procedures such as hyperparameter tuning, or others that involve a partial reinvention of the encoder of the model.

1.3.2 Specific Objectives

1. To train and evaluate the benchmark model using the different hyperparameter configurations selected for the present contribution.

2. To compare the performance of the different alternative models arising from the use of different convolutional models to replace the original encoder of the studied architecture.

3. To compare the feasibility and performance obtained by the different attentional encoders used to replace the original choice in the benchmark architecture.

Chapter 2

Theoretical Framework

Here is a summary of all the Deep Learning concepts that are scattered throughout this work. The different topics are covered according to their chronology and complexity, starting from basic neural networks and ending with fully attentional models. Each of the architectures are paired with their corresponding graphical and mathematical representations.

2.1 Artificial Neural Networks (ANN)

In the race to find a way to endow a computer with the inherent learning capacity of a human being, scientists are inspired by the anatomy of the human brain to come up with an innovative approach that marks a turning point in the area of artificial intelligence and machine learning [17].

Building on the initial work of McCulloch and Pitts, scientist Frank Rosenblatt refines a mathematical recreation of a human neuron capable of solving binary classification tasks, even modeling logical AND, OR gates [17]. This invention, known as **perceptron**, results in the beginning of the area known today as Deep Learning.

As shown in Fig. 2.1, the best known version of the perceptron today, performs an internal summation of the input arguments or features that make up the instance to be processed. Subsequently, the output is subjected to an activation function, finally obtaining the model prediction [18]. Unlike the initial approach of McCulloch and Pitts, Rosenblatt's contribution avoids having to modify the weights of the model by hand, but rather proposes the first mathematical methods to carry out this task with respect to the model output.

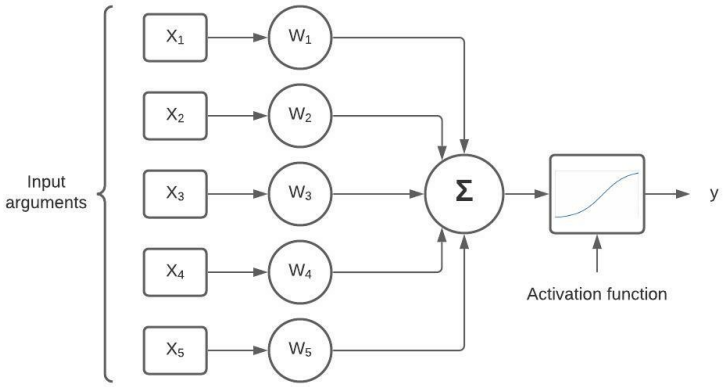


Figure 2.1: Overview of the current perceptron architecture.

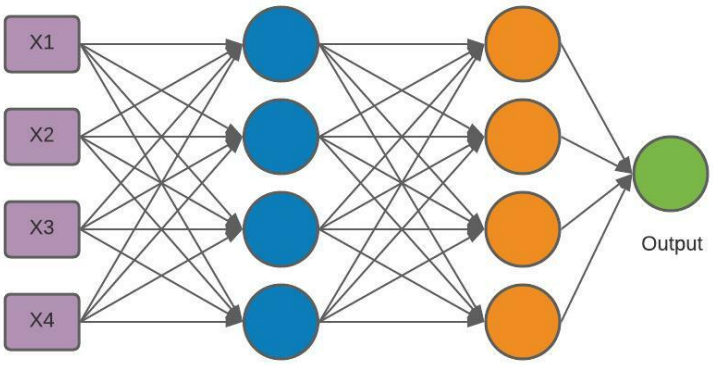


Figure 2.2: Overview of a multi-layer perceptron architecture.

However, in 1969, Minsky and Papert demonstrated the shortcomings of a simple neural network consisting of a single neuron. These limitations are exhibited mainly by demonstrating the inability to obtain a model capable of learning the XOR function, since it is not a linearly separable problem [19]. Precisely in this same work, these authors propose the idea of creating networks composed of multiple perceptrons organized in layers, thus being able to learn much more complex patterns.

Thanks to the latter authors, the neural architecture known as multi-layer perceptron (MLP) was consolidated. In general, this structure consists of three sections. On the one hand, a layer of input neurons, which only provide the input arguments to the network. On the other hand, at the end of it is the output layer, whose purpose will be to carry out the last computations necessary to generate the final prediction of the model. Finally, in the intermediate zone of the architecture are the hidden layers [18]. These layers provide the

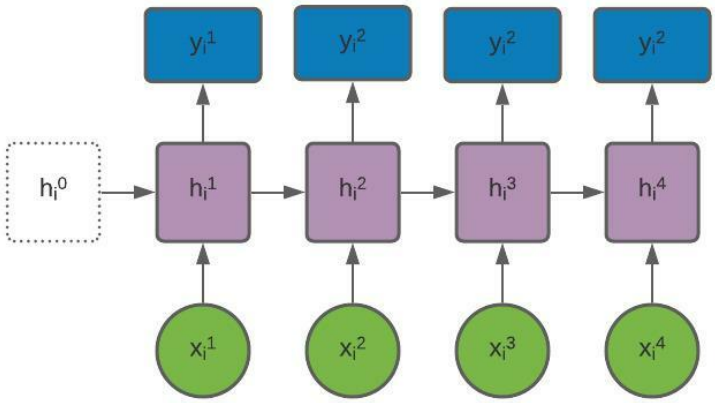


Figure 2.3: Overview of a recurrent neural network with a single hidden layer.

largest amount of computation and learnable parameters within the network. The number of layers within this portion of the model can be increased depending on the problem to be solved, and architectures with a considerable number of these intermediate layers are called **deep neural network**.

For the example shown in Fig. 2.2, taking all the parameters involved to the matrix plot, the model output would be expressed as:

$$y = W_y f(W_2 f(W_1 x)) \tag{2.1}$$

where W_1, W_2 and W_y correspond to the matrices formed by the weights of the corresponding neural layers, while f corresponds to an arbitrary activation function [18].

2.2 Recurrent Neural Networks (RNN)

Despite the feasibility of MLP for learning more complex patterns, the reality is that its nature makes it impossible to process sequential data. As seen in its mathematical formulation, each processed instance is completely independent, since no prior information is considered for the generation of subsequent predictions. As a result, data such as frames of a video, letters within a word or vital signs represent a new challenge that cannot be properly addressed by the previously introduced methodology.

Thus, **recurrent neural networks (RNNs)** appear in order to work with sequen-

tial data [20]. For this new approach, the preservation of information between iterations is proposed. Starting from the example shown in Fig. 2.3, an abstraction of a neural architecture composed of a single layer of hidden neurons is considered.

For this new scenario, the examples that compose each input instance must be processed individually in different time steps. What is intended is that at each iteration, when computing the output of the hidden neurons using the formula used for the MLP, the output generated by these same neurons in the previous time slot is also incorporated.

Explicitly, this methodology is expressed as follows:

$$h_i^t = f(W_h x_i^t + U h_i^{t-1}) \tag{2.2}$$

where h_i^t corresponds to the current hidden state of the example i; h_i^{t-1} represents the previous hidden state; and U appears as a new matrix of learnable weights in charge of directly modifying the information to be reincorporated into the new time space [20].

Once the hidden layer output is generated, one can proceed with the same methodology used to compute the network prediction as would be done in an MLP architecture.

$$y_i^t = f(W_y h_i^t) \tag{2.3}$$

2.3 Long Short Term Memory (LSTM)

The mathematical approach behind the RNN's manages to establish a way to implement a memory system over different time spans, constantly reincorporating the previous hidden state. However, the new inherent problem of this new approach lies in its shortcomings in having only short-term memory.

Due to their design, RNNs tend to progressively forget the content of the initial hidden states as the number of computational time slots increases within the architecture. For example, in the case of needing to process a 40-word text, the hidden state generated from the first element of that input will be entirely reconsidered for the generation of the second word. However, as new hidden states are computed, the information related to the first word of the text will be considerably attenuated for the time slot $t = 40$, which is problematic due to the nature of the natural language processing (NLP) area.

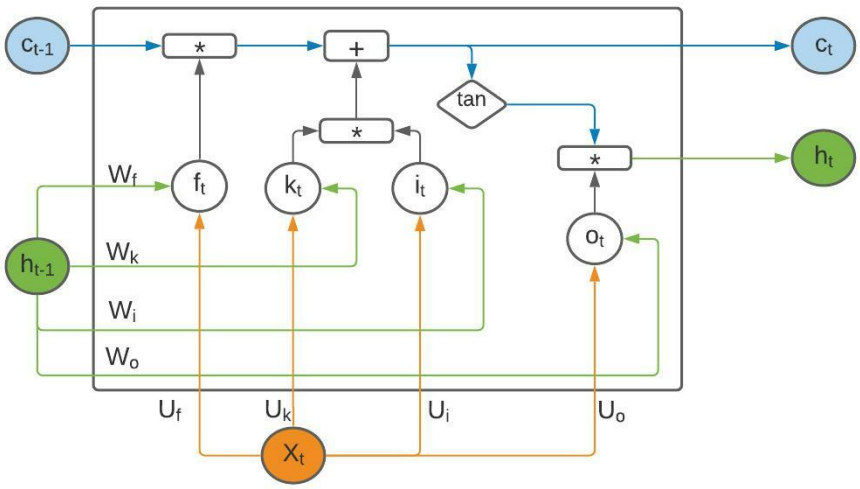


Figure 2.4: General representation of the composition of an LSTM unit, with its inherent gates and learnable parameters.

Thus, the architecture known as **long short-term memory (LSTM)** was born [21]. In this variation of the recurrent network, the aim is to continue incorporating the previous hidden state in each time instance. However, the introduction of new elements within the operation of an LSTM layer is noteworthy. On the one hand, the idea of using a memory system is consolidated, relegating this role to an argument known as **cell-state**. In this way, this new component will seek to function as a much more extended and selective memory that will interact directly with the corresponding hidden state.

Speaking of its explicit operation, this architecture follows the same line of classical RNN's, generating a computation relating the input vector and the hidden state of the previous spacetime in a linear transformation. At the same time, this result is subjected to an activation function, which by convention uses a hyperbolic tangent [21].

$$k_t = tanh(X_t U_k + h_{t-1} W_k) \tag{2.4}$$

Now, both for the computation of the current hidden state and the corresponding cell state, the LSTM relies on the incorporation of three gates in charge of very well defined functions: input, output and forget gate.

These gates are MLP structures that involve again the input vector X_t and the previous hidden state h_{t-1} within an activation function. However, it should be emphasized that

each of these gates will incorporate a different weight matrix that will proceed to directly modify the hidden state to be processed.

$$f_t = s(X_t U_f + h_{t-1} W_f) \tag{2.5}$$

$$i_t = s(X_t U_i + h_{t-1} W_i) \tag{2.6}$$

$$O_t = s(X_t U_o + h_{t-1} W_o) \tag{2.7}$$

where f_t, i_t, and O_t are the outputs of the forget, input and output gates; having that (U_f, W_f), (U_i, W_i), and (U_o, W_o) are their respective learnable parameters applied, on the one hand, to the current input, and on the other hand, to the previous hidden state [22].

Once the outputs of the corresponding gates have been computed, we proceed to update the cell state, in order to generate the final output. Considering that the current hidden state of a classical RNN would correspond only to that computed in k_t, for this new approach, such output must first be considered for the update of the current cell state C_t as follows:

$$C_t = f_t C_{t-1} + i_t k_t \tag{2.8}$$

In this way, f_t is able to regulate the relevance that the content previously collected in the cell state will have for the next iteration. Likewise, input gate plays a similar role, this time modifying the impact of k_t within the long-term memory of the system, being even able to disregard such information in case the values of i_t tend to zero [21].

Finally, the network prediction equivalent to the current hidden state h_t will be computed from the previously updated cell state. After the application of an activation function, the output gate will directly modify the information captured in the cell state to consolidate the corresponding output [21].

$$h_t = O_t * tanh(C_t) \tag{2.9}$$

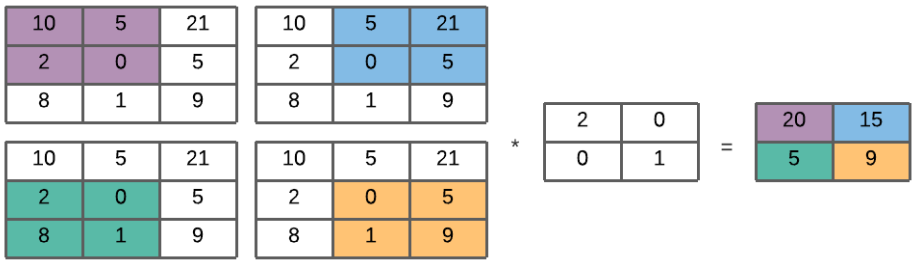


Figure 2.5: Example of application of convolutional processing to a 3x3 image using a 2x2 kernel and a 1 unit stride.

2.4 Convolutional Neural Networks (CNN)

Despite the feasibility of the **multi-layer perceptron** to perform image classification tasks, relevant information was neglected by these architectures during the processing of the corresponding instances. Since these models receive as input a one-dimensional vector of data, image processing was only feasible if the images were subjected to a flattening process. This previous procedure results in disregarding the spatial distribution of the pixels along the image, which is why an approach was sought to preserve this information in order to improve the corresponding state of the art.

Given this context, the idea arises to take advantage of **convolutions** as the main mechanism for image feature extraction.

This process seeks to generate a matrix such that each of its values contains relevant information of the processed image, seeking to respect the spatial distribution of the data. For this purpose, a convolution matrix (also known as **kernel**) is used, which will be superimposed on top of the image, going from left to right, and from top to bottom moving a certain number of pixels known as **stride**. In each iteration of this process, the values present in the region where the convolution matrix is located will be subjected to a linear combination with the kernel, thus obtaining the output pixel containing the analyzed information of the original image [23, 24]. As shown in Fig. 2.5, an example is carried out with a 2x2 kernel, a 3x3 image and a stride of 1, showing the correspondences between data by means of the colors shared between the input image and the resulting matrix.

As a second main component within the convolutional networks, the process known as **pooling** is incorporated. The objective of this process is the controlled and optimal

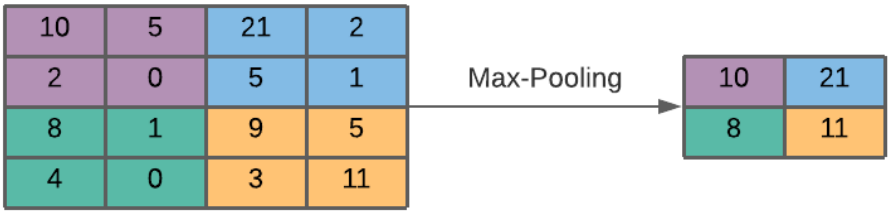


Figure 2.6: Example of application of max pooling process to a 4x4 image using a 2x2 kernel and a 2 unit stride.

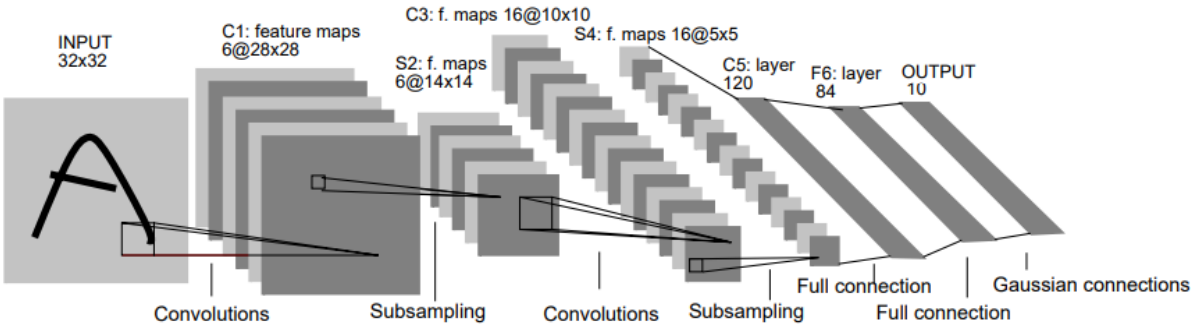


Figure 2.7: Graphical representation of the LeNet convolutional architecture [26].

reduction of the dimensionality of a given image by selecting or computing the values that best represent specific portions of the matrix. Similar to convolutions, a kernel of defined size will also be used during this process, which will be applied to sub-regions of the image until the entire image is covered. However, in this case the objective will not be to calculate each output pixel as a linear combination, but rather the kernel in question will apply a defined function using only the pixels of the image. Among the existing alternatives, the operation known as **max-pooling** stands out [25]. As shown in the example shown in Fig. 2.6, in each sub-region of the image, the kernel selects the pixel with the highest value so that it becomes part of the resulting matrix.

Using convolutions and pooling as tools for sequential image processing, the idea of using them in conjunction with MLP arose. One of the pioneer models in this field corresponded to LeNet [26], conceived for the classification of handwritten characters. As shown in Fig. 2.7, starting from a grayscale image, the authors organize the convolution and pooling processes as sequentially organized layers. The number of channels obtained increases progressively due to the application of multiple filters in each convolutional layer,

until fully connected layers are reached. In these final layers, all the features extracted in each of the corresponding channels are re-structured into a single vector, which can then be processed by the linear layers and generate the corresponding prediction.

2.5 Attention in Neural Networks

As the world of deep learning evolved, areas such as image segmentation and natural language processing opted to structure their neural architectures in such a way that they always consisted of two sections with complementary roles [27, 28]. On the one hand, a first encoder section is used to compact the input information, and then a decoder component takes advantage of this condensed feature representation in order to progressively construct an output that provides a solution to the task at hand.

The area of natural language processing was one of the most benefited from the trend of encoder-decoder architectures, with recurrent models being widely used as main players in one or both parts of the resulting neural networks [29, 30]. However, as the inherent memory problem of the recurrent approaches presented in previous sections was not completely solved, the demand for a solution to adequately process larger and larger input text compositions remained on the rise. Given these precedents, **attention mechanisms** arise as a way to provide the decoding part of a recurrent network with the ability to not only produce the corresponding output for a given time t, but also to perform a search for the most relevant elements of the supplied input in order to be taken into consideration for the computation of the corresponding hidden state [31].

While variations of the original attention system have taken place for different type of tasks [32, 33, 34], the reality is that they all start from the same original principle.

Starting from an encoding recurrent network, it receives an input text composed of its corresponding words $X = \{x_1, x_2, x_3, ..., x_m\}$. Instead of generating a single fixed-size element containing the full input expression, each of the hidden states h_j resulting from the encoding process, which will be used as annotation vectors to represent each input word. Using this approach, with respect to each hidden state of the decoder part s_i a $\beta i, j$ score associated with each annotation vector is computed. This task is assigned by convention to a single layer multi-layer perceptron model, formulated as follows:

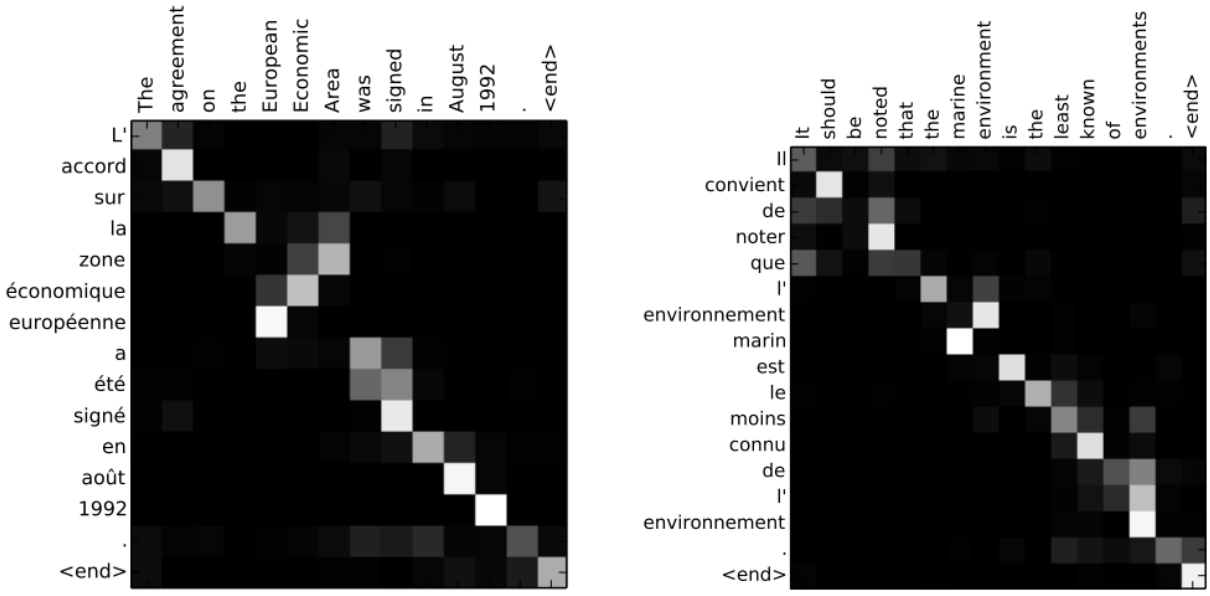


Figure 2.8: Attention matrices constructed from the weights associated with each input word with respect to the elements generated as output of the system [31].

$$\beta i, j = V * g(W * [h_j; s_i]) \tag{2.10}$$

where V and W are learnable parameters of the attention system, $[h_j; s_i]$ corresponds to the concatenation of the immiscible vectors, and g represents a nonlinear function [31].

After the calculation of each of the m scores, they are subjected to a softmax function in order to normalize each of the treated scores as weights $\alpha_{i,j}$:

$$\alpha_{i,j} = \frac{exp(\beta_{i,j})}{\sum_{j=1}^{m} exp(\beta_{i,j})} \tag{2.11}$$

In this way, each of these weights represents the relevance that each word of the input instance will have at the time of generalizing the token related to the hidden state s_i. By analyzing in a graphical way each of these weights $\alpha_{i,j}$, the matrices included in Fig. 2.8 are generated where each pixel corresponds to the weights computed using the attentional model. Having on the x-axis the words of the input sentence, and on the y-axis the output ones, we perceive with higher pixel intensities the input text positions of higher relevance for the generation of each output intensity.

Subsequently, a **context vector** is computed as a weighted sum between the attentional weights and their associated annotation vectors:

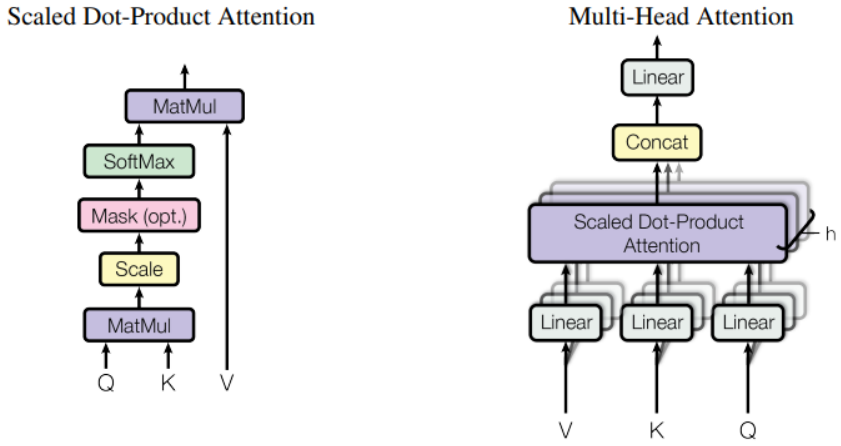


Figure 2.9: General composition of the Multi-head Attention and Scaled Dot Product blocks that make up the Transformer architecture [35].

$$c_i = \sum_{j=1}^{m} \alpha_{i,j} h_j \tag{2.12}$$

Precisely, it is through the context vector that the decoder can include the attentional information when computing the next hidden state s_{i+1}, satisfying the following expression:

$$s_{i+1} = f(Ux'_t + Ks_i + Qc_i) \tag{2.13}$$

where U, K and Q are learnable parameters, f a nonlinear function, and x'_t the input token to the decoder module for a specific time t [31].

2.6 Transformer Networks

Once the precedent created by the attention mechanisms was set, the initiative arose to formulate a system that dispensed with any convolutional or recurrent network and relied on attentional techniques for problem solving. This approach is embodied in the form of the innovative Transformer [35] network. This reimagining of machine translation systems stems from the hypothesis that suggests that attentional systems alone are sufficient to perform this type of task, dispensing with the recursive components that had historically been well accepted.

This end-to-end attentional architecture preserves the classic structure of an encoder-

decoder model, however, they use a generalization of the attention systems. A similarity with database information retrieval systems is used, where a query Q and key K define the search to be performed within a set of available values V, thus generating the desired result. This approach is materialized as the component known as **Scaled Doc-Product Attention**, which is diagrammed in Fig. 2.9. Moving this scheme to the mathematical terrain, this attentional principle would be embodied as:

$$Attention(Q, K, V) = \texttt{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.14}$$

where d_k is the embedding size selected to represent the words in the dataset. At the same time, the matrices V, K, and Q are the product of separate linear projections for each. In this way, the result of the softmax function generates a series of attentional weights similar to those seen in the classical method, which are applied on a representation of the input elements in order to obtain the context sought [35].

This Scaled Doc-Product Attention is employed within a composite structure called **Multi-Head Attention**. Referring again to Fig. 2.9, that component is responsible, in the first instance, for carrying out the projections for matrices V, K and Q, with the distinction that these structures are divided into h new matrices each. In this way, each trio of resulting matrices are subjected to the attentional process previously described, thus seeking to detect in a more detailed way a greater number of existing relations between the input words. The results obtained in each of these h heads are concatenated and finally subjected to a new linear projection.

Having already consolidated an attentional mechanism, it is incorporated as a cornerstone within the final architecture of the Transformer contained in Fig. 2.10. From this model it is highlighted that, unlike the recurrent models, the encoder of this new approach allows the processing of all the input text at the same time, instead of word by word, giving the possibility of parallelizing this part of the system. This is achieved thanks to the fact that, in addition to the use of classical learned embeddings from the area of natural language processing, the notion of **positional embeddings** is included. These are added to the classical embedding in order to provide the system with a way to keep a notion of the position of each word within the sentence, using sinusoidal and cosine functions as

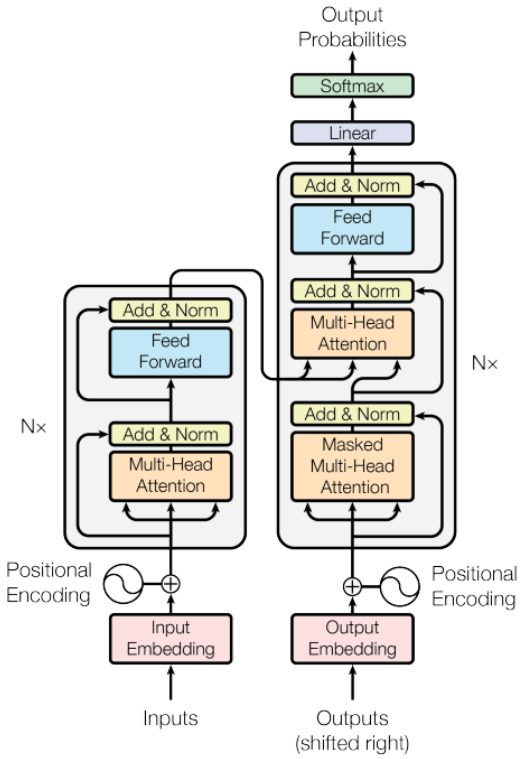


Figure 2.10: General scheme of the architecture known as Transformer, conceived to carry out natural language processing problems [35].

follows:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{2.15}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{2.16}$$

where pos represents the position of the token within the input text, i corresponds to the dimension, and d_{model} the embedding size [35].

Having already given a solution for the incorporation of positioning within sentences, we are now faced with the structure of the encoder part of the Transformer. This part of the model is composed of N identical coding blocks stacked one on top of the other. Each of these is made up of four functional blocks. First, once the V, K and Q matrices are generated, they pass through the Multi-Head Attention block previously analyzed in depth. After this, the addition of the content provided by a skip connection takes place, in conjunction with a layer normalization. Then, a double linear projection is carried out,

through a feed forward neural network, ending with a layer normalization preceded by the influence of a skip connection.

As for the blocks that make up the decoder, they present few important changes with respect to their encoding counterparts. Given that the model processes the entire text at the same time, during training it must be ensured that the decoder only uses as a premise the words prior to the time slot to be executed. In other words, the aim is to prevent the model from cheating by seeing the subsequent words, which is why a ground-truth masking is applied that varies as the translation process progresses.

Furthermore, as an intermediate sub-layer, the decoder block includes a new Multi-Head Attention module and another addition module with layer normalization. These maintain the same operation described so far, with the difference that at this specific point of the architecture, the knowledge of the encoder is incorporated, being now the one that provides the K and V matrices.

As in the encoder, the decoder blocks are stacked one on top of the other, ending the whole process in a final linear projection coupled to a softmax function, which will define the probability for each dictionary word to be selected as an inference at the time t of the translation.

Chapter 3

State of the Art

The following is an overview of the context and research direction of the technologies and areas of study involved in this work. Factors such as the most frequent datasets, metrics, hyperparameters and results of major relevance to the state of the art are taken into consideration.

According to the historical summary presented in Table 3.1, 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 [36]. 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 [31]. 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.* [37], and the one proposed by Ba *et al.* [38].

In order to substantiate the evolution within the area of image captioning, a brief historical review of relevant works is presented in Table 3.2. 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 [39]. Followed this research, the same author incorporated recurrent structures within an encoder-decoder model, a common factor among image captioning proposals [40]. 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.* [41], Vinyals *et al.* [42], and Donahue *et al.* [43] have reused this idea in their respective research efforts.

Finally, Table. 3.3 contains an excerpt from previous works that promote our hypothesis of incorporating a non-convolutional model within the proposed benchmark. The transformer architecture originates with the proposition that *attentional systems* are sufficient tools to replace approaches that employ recurrent networks for machine translation tasks. 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 [35].

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 [44].

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. By exploiting the decoder modules of the transformer we obtain the GPT architecture, whose later versions leave a hegemony mainly in text generation [45]. In contrast, 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 [46].

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% [47].

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 [48].

Subsequently, because of the considerable impact caused by the work "*An image is worth 16x16 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 [49].

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 [50]. This proposal was tested in image captioning tasks on the 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 [51]. 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 [8]. 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 [52].

Table 3.1: 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 3.2: 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 3.3: 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)

Chapter 4

Methodology

This section details all the aspects related to the execution of the experiments that will allow fulfilling the objectives of this work, together with the composition and formats followed by the dataset used. Additionally, specific details about the mathematical principles involved in the benchmark architecture, hyperparameters of major relevance, and auxiliary information about the nature of the alternative convolutional and non-convolutional models used for the respective experimental scenarios are shared.

4.1 Description of the Problem

4.1.1 The Dataset Structure

The dataset used for training the network is the 2014 version of the MS COCO variant oriented to image captioning tasks [53]. 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.

4.1.2 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 [54], respectively. Each image in the

Figure 4.1: 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`".

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

Table 4.1: Mapping system used to encode the caption the example image.

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.

4.1.3 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.

Considering that the model can work with descriptions of a maximum length of 52 words, the inclusion of three special characters within the mapping file is required. The network requires a start and end signal to delimit the extension of the descriptions. Also, since not all descriptions occupy the maximum sentence size, it is required to fill in the missing spaces within the encoded captioning with a character denoting a blank space.

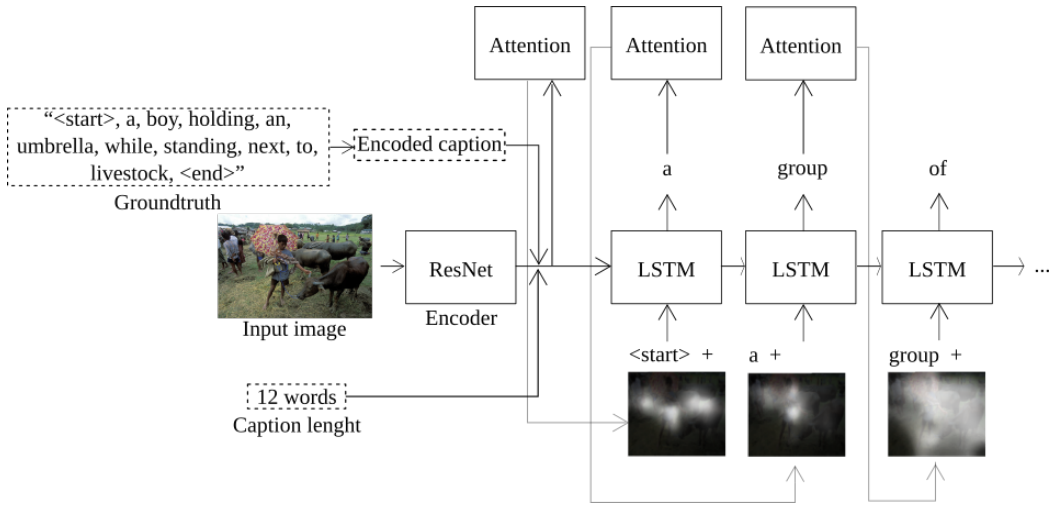


Figure 4.2: Overall representation of the convolutional encoder-decoder architecture built to generate real captioning. The model uses a pre-trained ResNet architecture as the encoder backbone, along with recurrent LSTM operations for the decoder. The objects with discontinuous contours are only used during the training stage.

As an example, in Fig. 4.1 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 4.1.

4.1.4 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.

4.2 Benchmark Model

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. 4.2, where an instance of the dataset is outlined in order to show its operation. The encoder makes use of transfer learning by borrowing the convolutional architecture of Resnet [55]. 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.

$$a = \{a_1, a_2, ..., a_L\}, a_L \in \mathbb{R}^D \tag{4.1}$$

On the decoder side, given the sequential nature of the problem to be solved, an LSTM recursive architecture is constructed [21]. 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.

$$\lambda_{ti} = Att(a_i, h_{t-1}) \tag{4.2}$$

$$\alpha_{ti} = \frac{exp(\lambda_{ti})}{\sum_{k=1}^{L} exp(\lambda_{tk})} \tag{4.3}$$

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.

$$\hat{z}_t = \sum_{i=1}^{L} \alpha_{ti} a_i \tag{4.4}$$

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

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto exp(\mathbf{L}_0(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t)) \tag{4.5}$$

where $\mathbf{L}_0$, $\mathbf{L}_h$, $\mathbf{L}_z$, and $\mathbf{E}$ are learnable parameters initialized randomly [8]. The objects in Fig. 4.2 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.

4.3 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.

4.3.1 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 $J(\boldsymbol{\theta})$ [56].

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:

$$\begin{aligned} J(\boldsymbol{\theta}) &= -\frac{1}{m}\sum_{i=1}^{m} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= -\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{Y}^{(i)} log\Big(\hat{p}^{(i)}\Big), \end{aligned} \tag{4.6}$$

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 $\mathbf{C}$.

4.3.2 Adaptive moment optimizer

In order to optimize our attention model through a gradient-based optimization method, we express the gradient vector of (4.6) with respect to θ as

$$\begin{aligned} \mathbf{g} &= \nabla_\theta J(\boldsymbol{\theta}) \\ &= \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{\theta}) \\ &= \frac{1}{m}\sum_{i=1}^{m} \left(\hat{p}^{(i)} - \boldsymbol{Y}^{(i)}\right)\boldsymbol{X}^{(i)}. \end{aligned} \tag{4.7}$$

To locate the minimum of $J(\boldsymbol{\theta})$, the proposed optimization algorithm moves to the negative direction of (4.7) iteratively. Our model computes individual adaptive learning rates for different parameters from estimates of first and second moments of $\mathbf{g}$ [57].

4.4 Experimental Setup

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.

4.4.1 Hyperparameter 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

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:

$$\boldsymbol{c_0} = f_{init,c_0}(\tfrac{1}{L}\textstyle\sum_{i=1}^{L} a_i)$$
$$\boldsymbol{h_0} = f_{init,h_0}(\tfrac{1}{L}\textstyle\sum_{i=1}^{L} a_i)$$

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 $\mathbb{B} = \left\{\left[\boldsymbol{X}^{(1)} : \boldsymbol{Y}^{(1)}\right], \ldots, \left[\boldsymbol{X}^{(m'_{tr})} : \boldsymbol{Y}^{(m'_{tr})}\right]\right\}$

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.

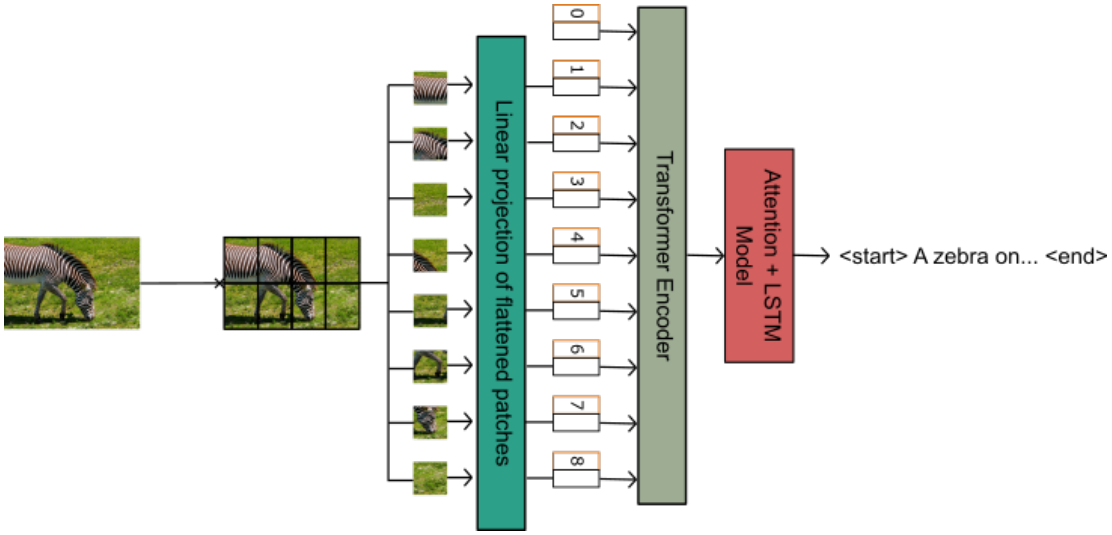


Figure 4.3: Overall representation of the ViT adaptation proposal for solving images captioning tasks. The MLP of the original implementation is replaced by the decoder used in previous experimental scenes.

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.

4.4.2 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.

4.4.3 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. 4.3.

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.

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.

Chapter 5

Results and Discussion

From Table 5.1, 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.

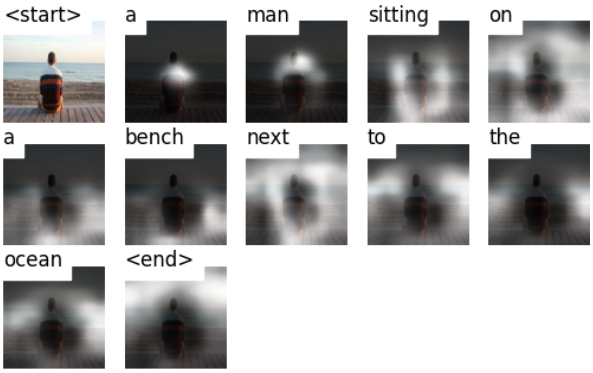
In addition to the quantitative results, Fig. 5.1 illustrates a captioning example generated using each one of the loss functions under study. The outcomes prove that the cross-entropy loss function is positioned not only as the one with the best results, but also the only loss function capable of generating 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 5.2 reveal a

	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

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

Table 5.1: Experimental results using *Top-5* accuracy and the *BLEU-4* performance metric for each one of the loss functions under study.



(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.

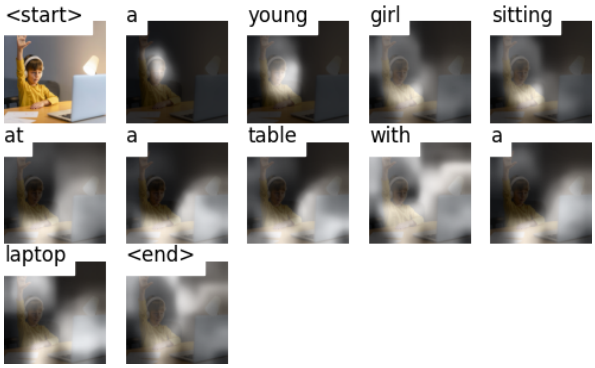
Figure 5.1: Image captioning results using an attention model with: (a) cross entropy loss, (b) MSE loss, (c) NLL loss, and (d) KLDIVLOSS. The results reveal an inadequate inference of MSE, NLL and KLDIVLOSS functions. By far, cross entropy is the only loss function that allows a proper training of our attention model.

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. 5.2. 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.

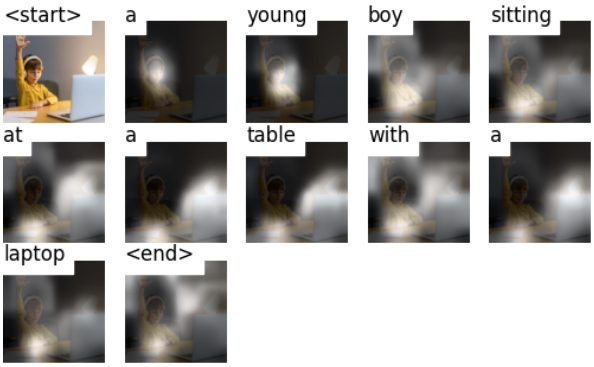
It is worth highlighting the performance of the RMSprop, which ranks as the third-best alternative, presenting a loss value of 3.663, along with 71.444 and 19.20 for Top-5 accuracy and BLEU-4, respectively. RMSprop shows promising results when comparing the output caption with the example image shown in Fig. 5.2. 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.

(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 5.2: 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).

	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

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

Table 5.2: Experimental results using the training loss, the *Top-5* Accuracy, and the *BLEU-4* performance metrics for each one of the optimizers under study.

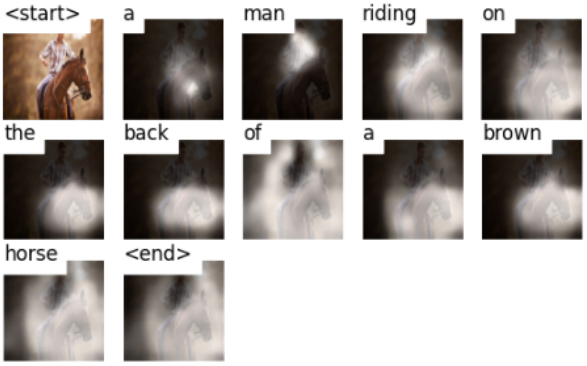
	BLEU-4	Top-5 Accuracy	Loss	Total Parameters	Training Time (Hours)	Inference Time (Seconds)	Computational Performance (GMAC's)
ResNet-101	20.10	73.092	3.413	42,500,160	5.6046	0.10765	7.85
ResNet-152	**20.20**	73.077	3.412	58,143,808	6.4077	0.14021	11.58
VGG-16	20.00	73.069	3.413	14,714,688	4.8353	0.08430	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**

Table 5.3: 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.

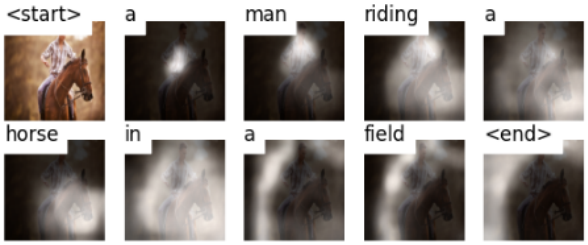
Finally, the SGD and Adadelta optimizers provided the worst results. Although both optimizers presented slightly different metrics, it is observed that neither of them were able to create a model capable of generating meaningful captions.

Now, referring to the results of the encoder testing phase shown in Table 5.3, two isolated analyses were conducted. At first, when looking for the convolutional model that 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. In contrast, the situation 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. 5.3. The ability of each of the models to 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.



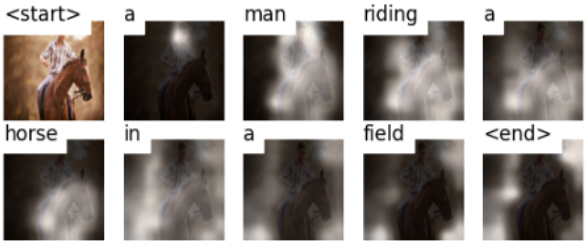
(a) Image captioning result using ResNet-152 as encoder.



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

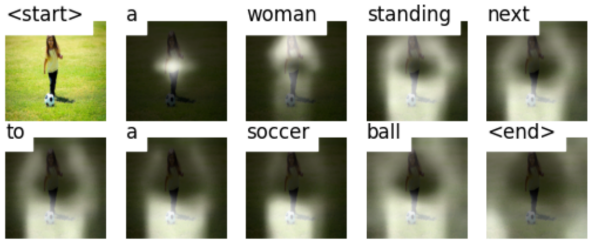
(c) Image captioning result using VGG-16 as encoder.

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

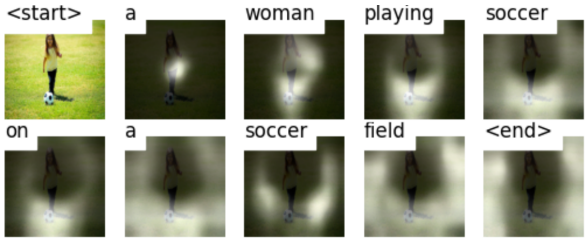


(e) Image captioning result using MobileNet V.3 as encoder.

Figure 5.3: 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.



(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 5.4: 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.

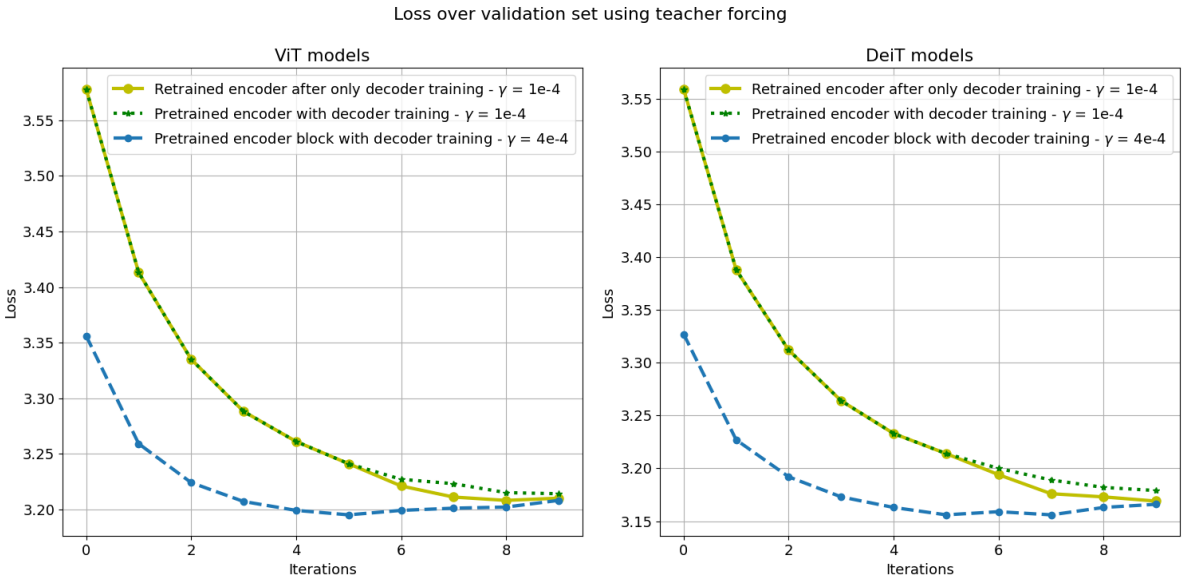


Figure 5.5: Evolution of the loss obtained during each of the corresponding iterations. These results were recovered using TF during the inference process on the validation set.

Relying on a second example, Fig. 5.4 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. 5.5 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 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

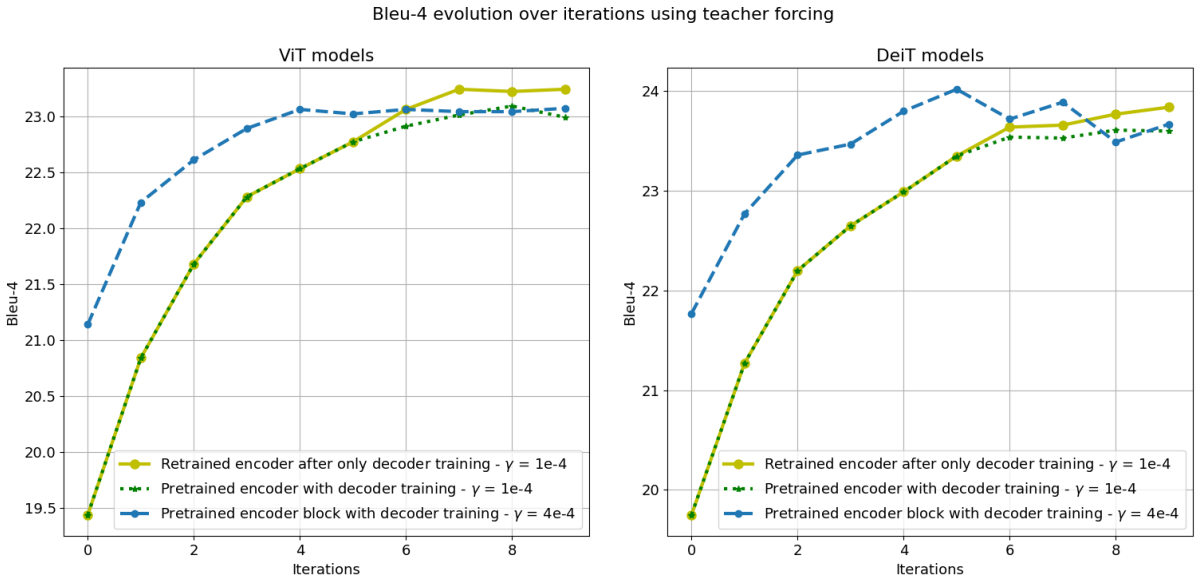


Figure 5.6: Evolution of the BLEU-4 metric obtained during each of the corresponding iterations. These results were recovered using TF during the inference process on the validation set.

similar visualization now using a metric related to the nature of natural language. Thus, Fig. 5.6 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. 5.7.

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 with a slightly higher γ in a very few number of iterations. Moreover, when using these results,

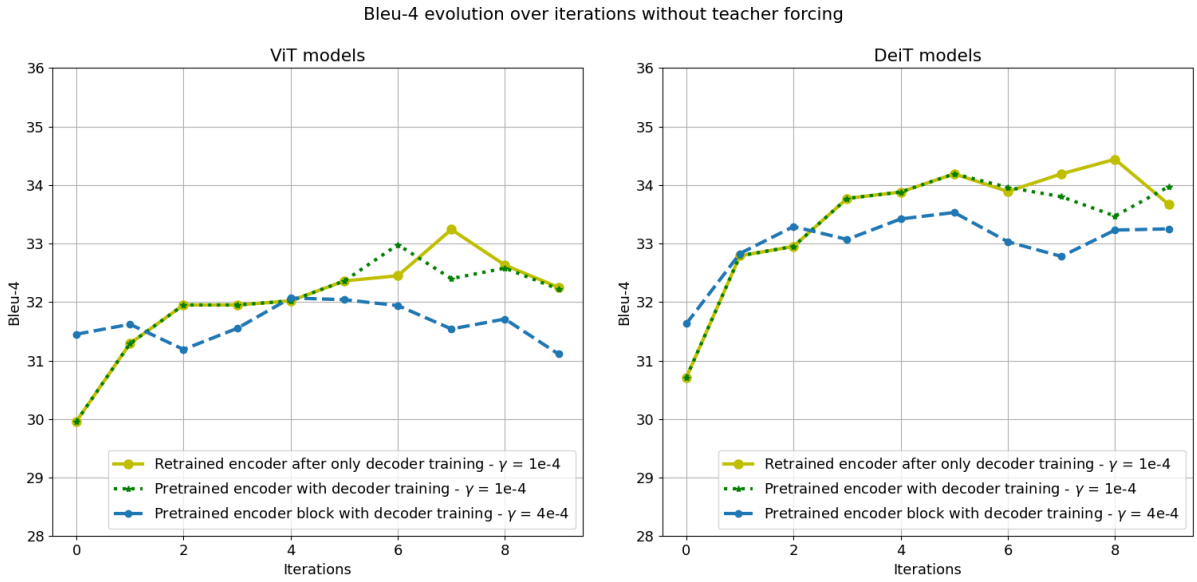


Figure 5.7: Evolution of the BLEU-4 metric obtained during each of the corresponding iterations. These results were retrieved without using TF during the inference process on the validation set.

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

Table 5.4: 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 units.

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 5.4 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.

Figure 5.8: 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.

As a complement to the quantitative results shown above, Fig. 5.8 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 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. 5.8, 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 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.

Chapter 6

Conclusions

1. 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 contrary, 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.

2. 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. First, 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. Then, 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.

3. Regarding the study involving the use of transformer-based architectures as a replace-

ment 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.

6.1 Future Works

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.), enabling an in-depth research of the attention architecture.

Bibliography

[1] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[3] Y. Yang, C. Teo, H. Daumé III, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 444–454.

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

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

[6] A. Rastogi, R. Arora, and S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic," in *2015 2nd international conference on signal processing and integrated networks (SPIN)*. IEEE, 2015, pp. 500–505.

[7] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[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," 2016.

[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 Systems with Applications*, vol. 165, p. 113967, 2021. [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 and Communication Technologies*, J. P. Salgado Guerrero, J. Chicaiza Espinosa, M. Cerrada Lozada, and S. Berrezueta-Guzman, Eds. Cham: Springer International Publishing, 2021, pp. 172–183.

[11] A. Byerly, T. Kalganova, and I. Dear, "No routing needed between capsules," *Neurocomputing*, vol. 463, pp. 545–553, 2021.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[13] Q. Wu, C. Shen, P. Wang, A. Dick, and A. v. d. Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1367–1381, 2018.

[14] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2407–2415.

[15] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Comput. Surv.*, vol. 51, no. 6, feb 2019. [Online]. Available: https://doi.org/10.1145/3295748

[16] H. Sharma, M. Agrahari, S. K. Singh, M. Firoj, and R. K. Mishra, "Image captioning: a comprehensive survey," in *2020 International Conference on Power Electronics &*

IoT Applications in Renewable Energy and its Control (PARC). IEEE, 2020, pp. 325–328.

[17] A. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[18] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, 07 2009.

[19] M. Minsky and S. Papert, "Perceptron: an introduction to computational geometry," 1969.

[20] M. I. Jordan, "Chapter 25 - serial order: A parallel distributed processing approach," in *Neural-Network Models of Cognition*, ser. Advances in Psychology, J. W. Donahoe and V. Packard Dorsel, Eds. North-Holland, 1997, vol. 121, pp. 471–495. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166411597801112

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2015.

[23] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.

[24] P. Kim, "Convolutional neural network," in *MATLAB deep learning*. Springer, 2017, pp. 121–147.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[28] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf

[30] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014.

[32] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," *arXiv preprint arXiv:1702.00887*, 2017.

[33] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[34] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523122100477X

[35] 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 *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran

Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[36] H. Larochelle and G. Hinton, "Learning to combine foveal glimpses with a third-order boltzmann machine," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 1. Curran Associates, Inc., 01 2010, pp. 1243–1251.

[37] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," 2014.

[38] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," 2015.

[39] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *International conference on machine learning*. PMLR, 2014, pp. 595–603.

[40] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014.

[41] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," 2014.

[42] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[43] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

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

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

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

[47] 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, 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/12/1939

[48] A. Patel and A. Varier, "Hyperparameter analysis for image captioning," 2020.

[49] 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 16x16 words: Transformers for image recognition at scale," 2021.

[50] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "Cptr: Full transformer network for image captioning," 2021.

[51] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 041–13 049.

[52] 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. 1s, mar 2021. [Online]. Available: https://doi.org/10.1145/3422668

[53] 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 – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

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

[56] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning to improve multi-hop searching and extended wireless reachability in v2x," *IEEE Communications Letters*, vol. 24, no. 7, pp. 1477–1481, 2020.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, 2015.