Remote Sensing Image Captioning with Pre-Trained Transformer Models

Sergio Zaera Mata

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Bruno Emanuel da Graça Martins

Examination Committee

Chairperson: Prof. João Manuel de Freitas Xavier

Advisor: Prof. Bruno Emanuel Da Graça Martins

Committee: Prof. Jacinto Paulo Simões Estima

May 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Abstract

Remote sensing images, and the unique properties that characterize them, are attracting increased attention from computer vision researchers, largely due to their many possible applications. The area of computer vision for remote sensing has effectively seen many recent advances, e.g. in tasks such as object detection or scene classification.

Recent work in the area has also addressed the task of generating a natural language description of a given remote sensing image, effectively combining techniques from both natural language processing and computer vision. Despite some previously published results, there nonetheless are still many limitations and possibilities for improvement. It remains challenging to generate text that is fluid and linguistically rich, while maintaining semantic consistency and good discrimination ability about the objects and visual patterns that should be described. The previous proposals that have come closest to achieving the goals of remote sensing image captioning have used neural encoder-decoder architectures, often including specialized attention mechanisms to help the system in integrating the most relevant visual features while generating the textual descriptions.

Taking previous work into consideration, this work proposes a new approach for remote sensing image captioning, using an encoder-decoder model based on the Transformer architecture, and where both the encoder and the decoder are based on components from a pre-existing model that was already trained with large amounts of data.

Experiments were carried out using the three main datasets that exist for assessing remote sensing image captioning methods, respectively the Sydney-captions, the UCM-captions, and the RSICD datasets. The results show improvements over some previous proposals, although particularly on the larger RSICD dataset they are still far from the current state-of-art methods. A careful analysis of the results also points to some limitations in the current evaluation methodology, mostly based on automated n-gram overlap metrics such as BLEU or ROUGE.

Keywords: Encoder-Decoder Models, Deep Learning, CLIP for Image Captioning, Remote Sensing Imagery, Remote Sensing Image Captioning.

Resumo

Conjuntos de imagens obtidas por detecção remota, e as propriedades únicas que caracterizam estas imagens, estão cada vez mais atraindo a atenção de investigadores em visão computacional, em grande parte devido às suas muitas possíveis aplicações. A visão computacional sobre imagens obtidas por detecção remota tem visto efetivamente muitos avanços recentes, e.g. em tarefas como a detecção de objetos ou a classificação de cenas.

Alguns trabalhos recentes na área abordaram a tarefa de gerar descrições em linguagem natural para imagens obtidas por detecção remota, efetivamente combinando técnicas de processamento de linguagem natural e visão computacional. No entanto, apesar de alguns resultados publicados anteriormente, ainda existem muitas limitações e possibilidades de melhoria. Gerar descrições textuais fluidas e linguisticamente ricas, mantendo a consistência semântica e boa capacidade de discriminação sobre os objetos e padrões visuais que devem ser descritos, permanece um desafio. As propostas anteriores que chegaram mais perto de alcançar os objetivos para a legendagem de imagens obtidas por detecção remota usaram arquiteturas neurais do tipo codificador-descodificador, muitas vezes incluindo mecanismos especializados de atenção, por forma a ajudar os sistemas na integração das características visuais mais relevantes ao gerar as descrições textuais.

Levando em consideração trabalhos anteriores, este trabalho propõe uma nova abordagem para a legendagem de imagens obtidas por detecção remota, usando um modelo codificador-descodificador baseado na arquitetura Transformer, e onde tanto o codificador como o descodificador são baseados em componentes de um modelo pré-existente que foi já treinado com grandes quantidades de dados.

Foram realizadas experiências com os três principais conjuntos de dados existentes para avaliar métodos de legendagem de imagens obtidas por detecção remota, respectivamente os conjuntos de dados Sydney-captions, UCM-captions, e RSICD. Os resultados mostram melhorias em relação a algumas propostas anteriores, mas particularmente no conjunto de dados RSICD, que é o maior, os mesmos estão ainda longe do actual estado-da-arte. Uma análise cuidadosa dos resultados também aponta para algumas limitações na metodologia de avaliação atual, principalmente assente em métricas automáticas baseadas em sobreposições de n-gram, como as métricas BLEU ou ROUGE.

Keywords: Modelos Codificador-Descodificador, Aprendizagem Profunda, CLIP para Legendagem de Imagens, Imagens de Detecção Remota, Legendagem de Imagens de Detecção Remota.

Contents

List of Tables

List of Figures

Acronyms

[CLS] Classification token. 46

[EOS] End Of Sequence token. 46

[SOS] Start Of Sequence token. 46

AI Artificial Intelligence. xi, 8, 24, 33, 34

BLEU BiLingual Evaluation Substitute. v, 60, 61

BPE Byte Pair Encoding. 45

CIDEr Consensus-based Image Description Evaluation. xii, 60, 62, 63

CIFAR Canadian Institute For Advanced Research. 43

CLIP Contrastive Language-Image Pre-training. 43–49, 69, 70

CNN Convolutional Neural Network. 14–20, 26, 27, 31, 36, 44, 47, 48, 50

COCO Common Objects in Context. 49, 56

ConVIRT Contrastive VIsual Representation Learning from Text. 44

CPTR Caption TransformeR. xi, 47, 48

CSMLF Collective Semantic Metric Learning Framework. 41

DenseNet Dense Convolutional Network. xi, 19, 20, 41

DeVISE Deep Visual-Semantic Embedding Model. 44

EfficientNet Improving Accuracy and Efficiency through AutoML and Model Scaling. 27

FAIR Facebook AI Research. 44

FC Fully Connected. 41

FCN Fully Convolutional Neural Network. 15, 36

FFN Feed Forward Neural Network. 14

GELU Gaussian Error Linear Unit. 26

GPT Generative Pre-trained Transformer. 44, 48

GRU Gated Recurrent Units. xi, 12, 13

ICLR International Conference on Learning Representations. 26

ICMLM Image Conditioned Masked Language Modeling. 44

IDF Inverse Document Frequency. 62

LAM Label-Attention Mechanism. 41

LCS Longest Common Subsequence. 62

LN Layer Norm. 26

LSTM Long Short-Term Memory. xi, 12, 13, 19, 29, 31, 36, 41

LSVRC Large Scale Visual Recognition Challenge. 18

METEOR Metric for Translation Evaluation with Explicit Ordering. 60, 61

ML Machine learning. 41

MLP Multi-Layer Perceptron. 12, 14, 26

MSP Multi-Head Self-Service Layer. 26

NLP Natural Language Processing. 26, 32, 61, 69

NMSLib Non-Metric Space Library. 46

NN Neural Network. 12

NNR Neural Network Compression and Representation. 12

NOCAPS Novel Object Captioning At Scale. 49

PASCAL Pattern Analysis, Statistical Modelling and Computational Learning. 15

R-CNN Region Based Convolutional Neural Networks. 48

RefineNet Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. 15

ReLU Rectified Linear Unit. 14, 17

ResNet Residual Neural Network. 17–19, 23, 27, 41, 45

RMSProp Root Mean Square Propogation. 22

RNN Recurrent Neural Network. 31, 33, 36, 47, 50

ROUGE Recall-Oriented Understudy for Gisting Evaluation. v, 60, 62

RSICD Remote Sensing Image Captioning Dataset. iii, v, xii, 46, 57–59, 69

RTRMN Retrieval Topic Recurrent Memory Network. 41

SAT Show-Attend-and-Tell. 41

SCST Self-critical Sequence Training. 41

SD-RSIC Summarization Driven Deep Remote Sensing Image Captioning. 41

SEO Search Engine Optimization. 33

SIFT Scale-Invariant Feature Transform. 15

SM Spatial Maps. 41

TF Term frequency. 62

TF-IDF Term frequency – Inverse document frequency. 63

TL Transfer learning. 41

UC University California. 56, 57

UCM University California Merced. iii, v, xii, 56, 57

USGS United States Geological Survey. 57

VGG Visual Geometry Group Network. xi, 17, 18

VirTex Visual Representations from Textual Annotations. 44

ViT Vision Transformers. xi, 26, 27, 45, 46

VOC Visual Object Classes. 15

VRTMM Variational autoencoder Reinforcement learning-based Two-stage Multi-task learning Model. 41

Chapter 1

Introduction

In this first Section, the key points of the work will be presented, that is, the motivation that has led to assessing the idea of carrying out work on this subject will be explained, as well as the problems that are tried to be solved through this proposal.

The Research Statement will also be described, which is a summary of your research accomplishments and future direction and potential of your work, as well as the contributions made by our proposal and how it has been reached.

Finally, an outline will be made about the internal organization of the dissertation from this chapter, accompanied by a brief introduction or summary of each of the points, with the aim that all the information collected in this dissertation is easier to find and catalogue.

1.1 Motivation

Increasingly, the use of different deep learning methods for the detection, evaluation and subsequent analysis of remote sensing images has attracted great attention from both the scientific community and private companies specialized in certain sectors, who see these advances as a way to fight the competition, standing on the crest of innovation and therefore generating higher revenues or profits.

This has been seen mainly with the remarkable progress that has been made in tasks such as the classification of global scenes with greater precision, the detection and identification of objects or the classification at the pixel level for land cover mapping and semantic segmentation.

However, some studies [29] have been able to show that, although great progress in these tasks, not enough attention has been paid to the semantic gap problem, which is that it is still very difficult to describe the content of a remote sensing image with precise, concise and concrete sentences in natural, light and fluent language.

This problem is mainly due to the characteristics of these systems, since they passionately combine natural language processing and computer vision techniques, thus generating results at a linguistic level that are completely conditioned to the processing results of visual inputs and their correct ability to recognize objects and understand the spatial and complex relationships that exist between them, to be able to make a complete description of the scene. [30]

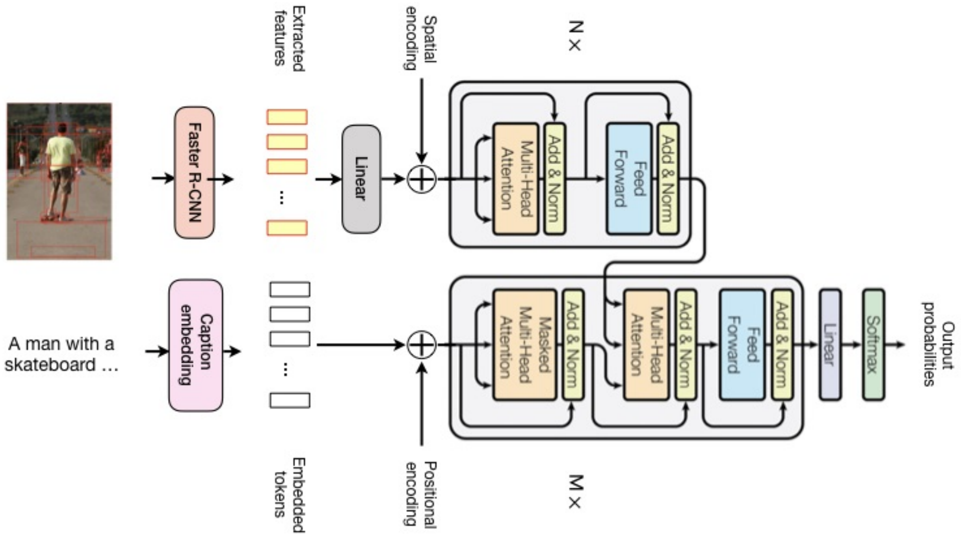


Figure 1.1: Trasformer-based encoder-decoder architecture for image captioning [2].

If the problem of the semantic gap can be satisfactorily solved and therefore concise descriptions of remote sensing scenes can be generated, the door would be opened to the possibility of using this new technology with all its potential to support numerous practical applications, which can cover fields as diverse as security, medicine or finance, with tasks such as retrieving images from textual queries, generating descriptions for scene classification, intelligence gathering in the field, among others.

Using the different cutting-edge approaches as inspiration, there are several ways to address the general problems of image captioning, but one of the most promising methods for researchers is the proposal of models based on deep neural networks.

In turn, the recent method proposals follow an encoder-decoder architecture that combines convolutional, recursive and attention components and uses models whose encoder and decoder components have already been previously trained by large volumes of data in their application area, either for image coding or for text coding [31].

The operation of these models is reduced schematically to an encoding component corresponding to a pretrained convolutional neural network that generates a representation of the image and a decoder based on recurrent neural networks generates the response word by word and in each step uses the attention to weight the different parts of the visual input according to the relevance of the token for the final prediction.

Although quite interesting results have been obtained using these structures, it is known that a significant improvement is possible concerning the composition of the aforementioned general architecture like the Transformer-based encoder-decoder model – see Figure 1.1. This project is a proposal that will serve to address the development and evaluation of innovative remote sensing image captioning methods, building on the prior state of the art [32].

1.2 Research Statement

In the future, we can expect a wide spread in the use of remote sensing images as a method of obtaining information, whether for Geo-spatial, military or commercial applications.

Through a full Transformer network, with fine-tuned components from a Contrastive Language–Image Pre-training (CLIP) model adjusted to the RSICD dataset for caption generation, called CLIP-RSICD, we will create a system that allows remote sensing image captioning to transform visual information into fluent and concise written information for subsequent analysis and evaluation.

1.3 Contributions

In summary, the main contributions of this project are the following:

- A novel approach to remote sensing image captions is proposed, a structure of Transformer-based encoder-decoder model, both the image encoder and the decoder, which consist of a Transformer model, use fine-tuned components from a pre-existing version of Contrastive Language–Image Pre-training (CLIP) model adjusted to the RSICD dataset for caption generation.
- We evaluate the impact that this new structure provides in-depth, as well as the peculiarities that make it unique and that allow the adjustment of both the encoder and the decoder, since when using the model pre-entering CLIP which is entering to perform a wide variety of tasks, which can be leveraged through natural language prompts to enable zero-trigger transfer to many existing datasets, enabling competitive performance compared to task-specific supervised models.

The neural architecture and all other code supporting the experiments builds on the HuggingFace Tranformers[1] library.

1.4 Organization of the Dissertation

To make an overview of how this dissertation have been organized, it should be noted that it follows the common guidelines for this type of work.

First Section 2 will try to review the fundamental concepts about the technologies, techniques and basic knowledge necessary to understand the rest of the work simply, as well as present the articles that laid the foundation to continue developing and advancing in this subject and therefore are directly related to the task developed.

On the other hand, Section 3 is based on the complete analysis of the task, where the specific approach proposed to solve the problem raised throughout the introduction is detailed, as well as all the mechanisms that have intervened in the process and that has served to achieve its development.

Next, Section 4 describes in-depth the set of data used during development, whether databases or external metadata, as well as the evaluation metrics used, to evaluate the results obtained after the entire process. It also incorporates the details of the implementation regarding the implementation, collection, extraction and storage of the resulting data, thus allowing discussion and therefore evaluation of the results obtained.

[1] https://huggingface.co/

Finally, Section 5 serves to conclude all the work, in this way it summarizes the main findings obtained, which will have already been discussed in the previous point, as well as informs about the new possible future routes on which the continuation could focus with the work.

Chapter 2

Background

2.1 Fundamental concepts

This chapter presents the fundamental concepts that will allow the understanding of the rest of the study, introducing the previous knowledge of the field of study, as well as referencing the articles from which all the information has been extracted.

In this first part, it is explained how little by little the technology of artificial intelligence appeared and more specifically Deep Learning, which is exactly how it works internally and how it has managed to integrate without realizing it into our daily lives. Once the general explanation is made, the chronological context is displayed and the fundamental concepts in terms of Deep Learning and Neural Networks are explored, which range from the simplest to the most complex, ending with one of the most important architectures the Transformer.

2.1.1 Learning with Deep Neural Networks

We are currently living in the era of big data, where all areas of science, industry and even entertainment generate massive amounts of data, this presents us with unprecedented challenges in terms of the analysis and interpretation of this data, as well as its classification and subsequent storage.

Given the current and unprecedented availability of vast amounts of data from computing resources, there is a great resurgence of interest in the application of data-driven machine learning methods for problems for which the development of solutions by engineering and conventional methods would be practically impossible [33]. For this reason, there is an urgent need for new machine learning and artificial intelligence methods that can help use this data [34].

The areas of application of these methods cover a large number of sectors, but the most important include: image recognition [35, 36], voice recognition [37], natural language understanding [38], acoustic modeling [39] and computational biology [40, 41, 42, 43].

Since Alan Turing 1950 wrote his article on the possibility of an electronic computer behaving intelligently through programming, thus recreating artificial intelligence, it has experienced rapid growth in research and development during the last decades – see Figure 2.1.

It is interesting to remember that one of the first people to work in this field and therefore one of the fathers of artificial intelligence was John McCarthy [44], who in the 1990s defined artificial intelligence as “the artificial intelligence is the science and engineering to make intelligent machines, especially intelligent computer programs ”.

This definition is based on the more practical concept of it, since the term ”AI” is used when a machine simulates functions that humans associate with other human minds, such as learning and problem-solving [45].

Artificial intelligence appears mainly intending to design and implement computer systems capable of solving problems that usually exceed the capacity of the human being, which usually correspond to high complexity and/or natural tasks [46].

An important characteristic of AI is its multidisciplinary. Artificial intelligence is highly linked to the information technology sector, but the development of AI projects also requires notions from logic, linguistics, cognitive sciences... due to the wide variety of fields of application it has, such as: medicine, industry, banking and finance, etc [47].

In this field of artificial intelligence, we can find two main blocks, Machine Learning and Deep Learning, which are necessary to explain to understand why the decision to use one and not the other in the development of the project.

While, in machine learning, an attempt is made to extract new knowledge from a large set of preprocessed data, previously loaded into the system, and programmers need to participate directly and actively formulating the rules of the machine so that it learns based on them, where it even forces human intervention to correct its mistakes.

In Deep Learning, it is possible to work with a much larger volume of data and the active intervention of the programmers is not necessary, since the training cycles are much longer, being the machine/system the one that is updated and learned in each cycle.

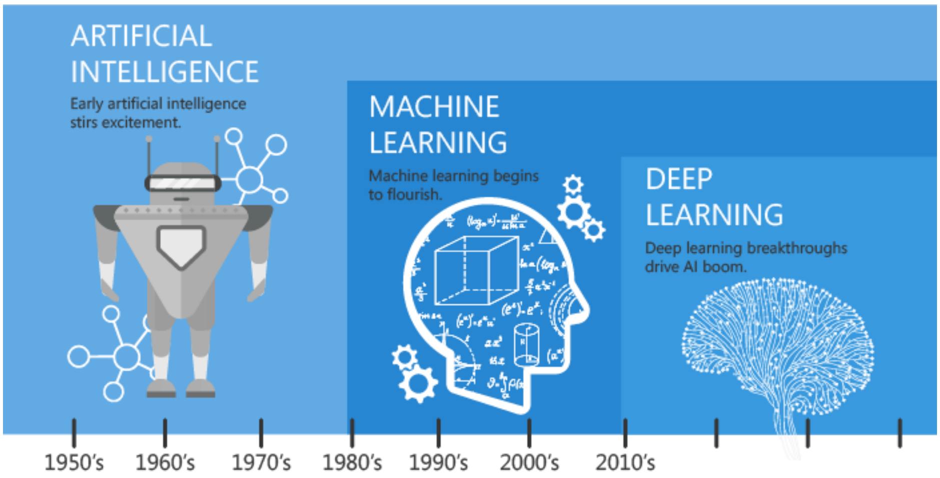

ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Figure 2.1: The evolution of artificial intelligence over the years [3].

Deep learning doesn't rely as much on the human experience as traditional machine learning, allowing us to make discoveries in data even when developers aren't sure what they're trying to find [48].

To finish this section, it is necessary to make a detailed explanation of the most basic operation of Deep Learning whose basic entity of any neural network is a neuron model, which can be observed in detail in Figure 2.2.

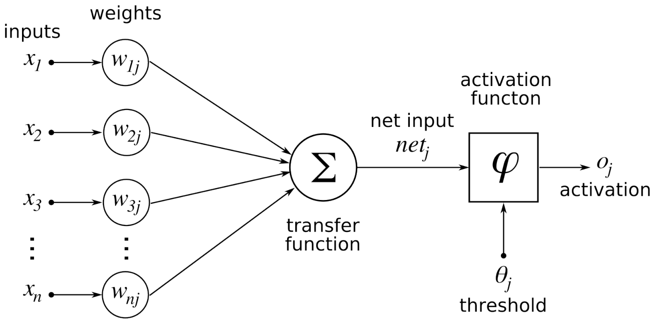


Figure 2.2: Artificial Neuron Model [4].

A basic neuron model is mainly characterized by consisting of input, x, along with a bias, b is weighted with, w, and then they are summarized together. The bias, b, is a scalar value while the input x and the weights w are vector values, that is, $x \in \mathbf{R}n$ and $w \in \mathbf{R}n$ with $n \in \mathbf{N}$ corresponding to the dimension of the input.

The sum of these terms, that is, $z = w^T x + b$ then forms the argument of an activation function, ϕ, which results in the output of the neuron model [49]:

$$y = \phi(z) = \phi(w^T x + b)$$

Once the model of a basic neuron has been described, we must know that to perform more complex tasks, larger networks are needed and therefore more neurons that connect.

This term is known as a neural network, which, as already mentioned, is made up of neurons connected, where each connection of our neural network is associated with a weight that dictates the importance of this relationship in the neuron when multiplied by the input value.

Each neuron has an activation function that defines the neuron's output, this function is used to introduce non-linearity in the modelling capabilities of the network and therefore there are a variety of activation functions with a variety of uses for each one.

These activation functions are mainly used to propagate the output of a neuron forward, where the output will be received by the neurons of the next layer to which this neuron is connected, serving as already mentioned to introduce non-linearity in the network modelling capabilities.

Some of the most common activation functions today can be observed in the following image, concluding that the shape determined by each function will directly affect the output of each of the neurons – see Figure 2.3.

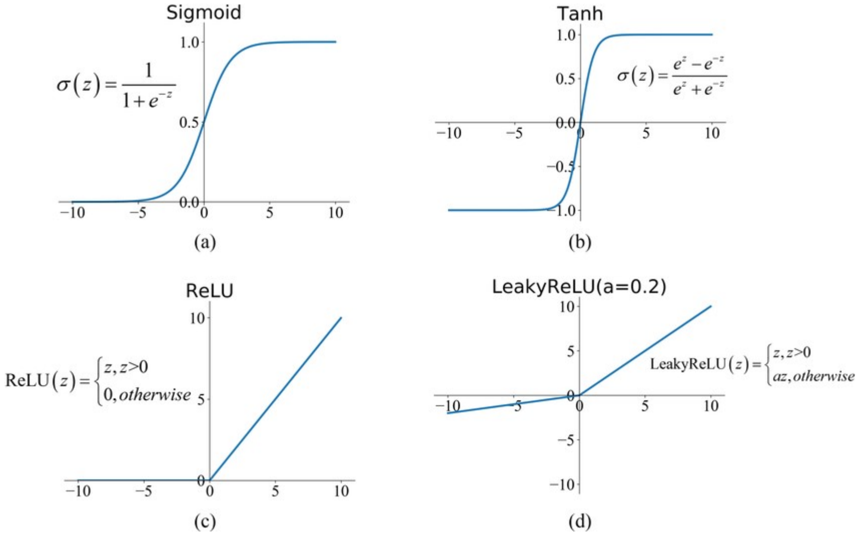


Figure 2.3: Commonly used activation functions [5].

The term ”train a neural network”, which has already been mentioned previously in this section, refers to the process by which the system learns the values of our parameters (weights wij and biases bj) and is one of the parts, but the most genuine part of Deep Learning.

This learning process in a neural network can be seen in a simplified way as an iterative ”round trip” exercise through the layers of neurons, where the action of ”going” is a forward propagation of information and the action of ”return” is a backward propagation of information.

This first phase of forwarding propagation occurs when the network is exposed to the training data and it moves through the entire neural network so that its predictions (labels) are calculated, in short, passing the input data through the network in such a way that all neurons apply their transformation to the information they receive from the previous layer and send it to the neurons of the next layer.

When the input data has already crossed all the layers and all of your neurons have performed their calculations, the final layer is reached with a label prediction result for those input examples.

Once the process is finished, a loss function is applied to estimate the error and thus compare and measure how good/bad our prediction result was about the correct result, where ideally, our cost would be zero, which would indicate that there is no divergence between the estimated and expected values, but that is a real case it would provide us with a value other than zero, which would vary depending on the effectiveness of the system.

Therefore, as the model is trained, the weights of the neurons' interconnections will gradually be adjusted until good predictions are obtained – see Figure 2.4.

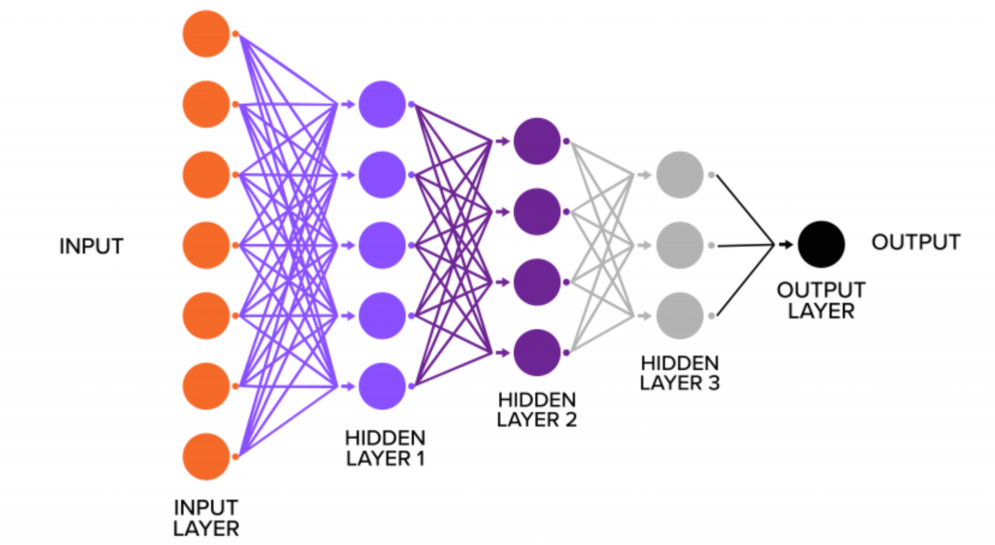


Figure 2.4: Deep Neural Network Model [6].

On the other hand, once the loss function is calculated, its information is propagated backwards from the output layer to all the neurons of the hidden layer that contribute directly to the output, although these will only receive a fraction of the total loss signal, depending entirely on the relative contribution that each neuron has contributed to the original output, this process is repeated layer by layer until all neurons in the network have received a loss signal.

Once this information is disseminated, it is time to adjust the weight of the connections between neurons, for this, we will use a technique called gradient descent, which changes the weights in small increments with the help of calculating the derivative of the loss function, which allows us to see in which direction to "descend" towards the global minimum, a process that will be repeated throughout all the iterations (epochs) using the entire dataset that we pass to the network in each iteration [50].

As we have seen, artificial intelligence and in particular deep neural networks can be found in a variety of domains [51], such as Proof of theorems, Natural language processing, Recognition and understanding of speech, Interpretation of images and vision, among others [52]. This great versatility that we can find in the fields of application, is applied in turn in the large number of different structures that exist, which constitute processes that are completely different from each other and that therefore their use can be reduced to single field research.

In this way, below we will make a brief description of the most relevant structures that separate the field, as well as their main characteristics, which differentiate them from each other and we will also comment on the key aspects and their possible best applications or uses in a real environment.

Feed Forward Neural Network (NN) - Also called a multilayer perceptron (Multi-Layer Perceptron (MLP)), it is the simplest architectural form of an NN, as it is primarily based on a feedback structure [53]. In this type of neural network, linear or non-linear activation functions can be used [54] and although it has been mentioned that its structure is based on feedback, it is important to note that there are no cycles/connections in the NN that allow direct feedback.

Finally, it is relevant to note that both in this network, as in the general case, the depth of a network denotes the number of non-linear transformations between the separation layers, while the dimension of the hidden layer, that is, the number of hidden neurons, determines the width of this.

Recurrent NN - This type of artificial neural network is used in speech recognition and natural language processing since its operation and learning are based mainly on the recognition of sequential characteristics and its subsequent use of patterns for the prediction of the most common scenario probable – see Figure 2.5.

This type of network is characterized by its use of feedback loops, which cyclically process a sequence of data, but allow the information to persist between each loop, creating an effect similar to that of memory.

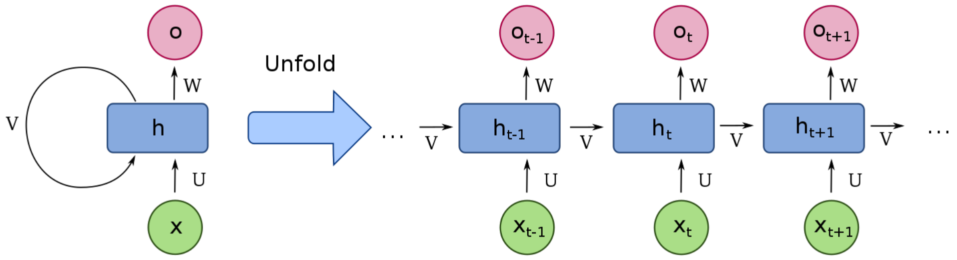


Figure 2.5: Recurrent Neural Network [7].

Within this type of neural network, we can find two main variations, Long short-term memory (LSTM), which were created by Hochreiter and Schmidhuber in 1997 [55] and the Gated Recurring Units (GRU).

These variations are different from normal Neural Network Compression and Representation (NNR)s and thanks to their new internal structure, they allow for managing long-term dependencies [56], as well as avoiding the problem of the disappearance of the gradient [57, 58], but there are small details that differentiate each of the versions. These differences are based on their execution capacity and memory retention, since while the GRUs execute faster because they require less memory and therefore fewer training parameters, the LSTMs are slower in their training process, but the result obtained is more precise in the dataset due to the use of a longer sequence.

Thus, concluding that for a large sequence or with a very critical precision objective, the use of an LSTM is chosen, while for lower memory consumption and a faster operation, the GRU is chosen – see Figure 2.6 [59].

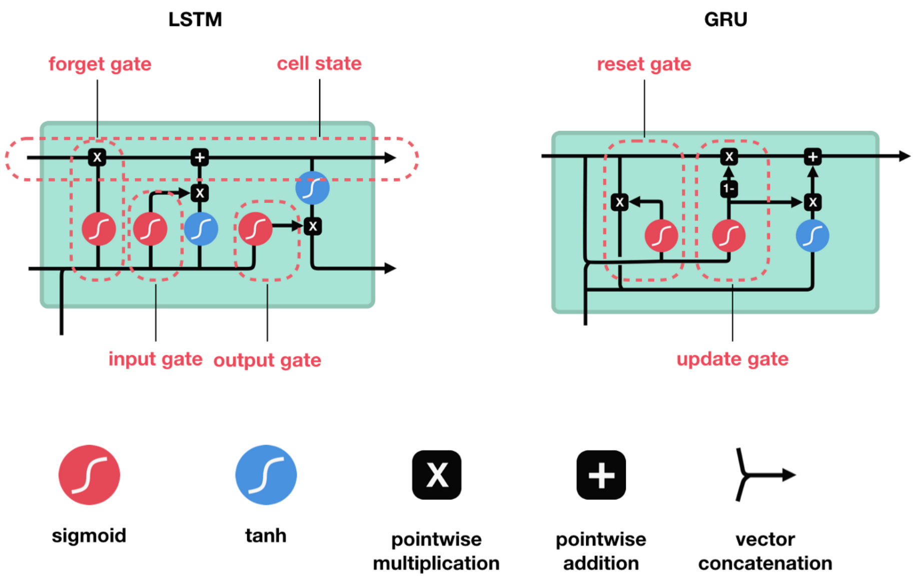


Figure 2.6: Comparison between LSTM and GRU models [8].

There are many types of artificial neural networks that work in different ways to achieve different results, the most important part of understanding neural networks is that their design is built to imitate how neurons in the brain work and in this way, as a result, the models can learn more and improve more with more data and more use.

This fact sets them apart from traditional machine learning algorithms that tend to plateau after a certain point, as neural networks can truly grow almost infinitely as the model receives more data and more use.

Below we can find Figure 2.7 which completes the explanation of the set of networks and reflects the reality of their internal structure, both of the networks previously explained in the section, and of more complex versions/variants of these.

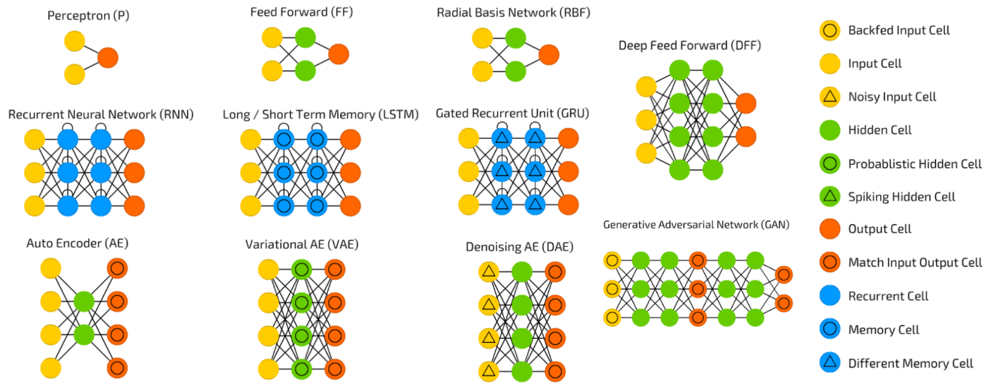


Figure 2.7: Different Types of Neural networks [9].

2.1.2 Neural Networks for Image Processing

As we have already seen in the previous section, there is a great variety of different neural networks, each one of them with its characteristics and uses, but in this section, we will focus mainly on convolutional neural networks (Convolutional Neural Network (CNN)), their unique properties and we will explain why they are so relevant to our case study – see Figure 2.8.

These types of networks are quite similar to Feedforward networks (Feed Forward Neural Network (FFN)), but with the special feature that more complex mechanisms are used such as convolutional layers, the Rectified Linear Unit (ReLU) function and the grouping method which form a network of much larger and more interconnected.

While in the most basic neural networks, the neurons of each layer are connected to all the neurons of the next, each connection being a parameter in the network, in CNN networks a local connectivity mechanism is used between the different neurons, where each neuron is only connected to the neurons close to this of the next layer, which not only implies a reduction in the number of total parameters but also allows the creation of groupings between different neurons, called nuclei.

Something important to note is that this type of network has been the main architecture used in tasks such as the recognition and classification of images to detect objects, recognize faces or the categorization of visual elements, since due to its internal structure described, it allows to group, with great ease, the similarities between different visual elements and later perform the recognition of these objects, being able to identify faces, street signs, animals, etc. [60].

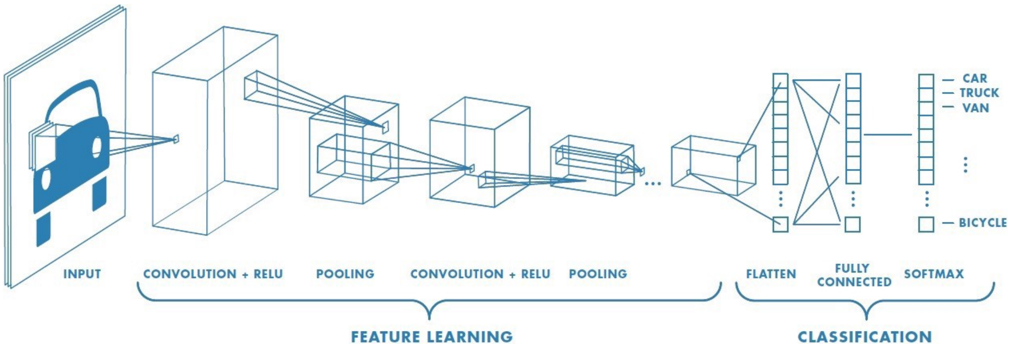


Figure 2.8: Convolutional Neural Network [10].

Although the concept of Convolutional Neural Networks may sound like a strange combination of biology, mathematics and computer science, as already mentioned, these networks have been the cause of some of the most relevant innovations in the field of vision, artificial intelligence and image processing, even comparing them as regularized versions of the Multilayer Perceptron (MLP), since they were developed based on the inner workings of neurons in the animal visual cortex [61].

A simple explanation can be given through the use of a simple example such as the Image Classification task, where the system must identify an element in the image and classify it, generating numbers that describe the probability that the image is of a certain class.

In this way, our brain, when observing an element such as a dog, with the task of identifying and classifying it, must analyze its most notable characteristics such as the number of legs, size or fur. Similarly, the model or system must perform a series of processes through which it can extract low-level features, such as edges and curves, and then build more abstract concepts through a series of convolutional layers and then, using the information obtained in the initial levels, must generate high-level features such as legs or eyes, with the ultimate goal of being able to identify the object or element.

As we have already mentioned, since its appearance, convolutional neural networks (CNN) have become the benchmark in computer vision tasks and have been one of the first models to outperform traditional methods in tasks such as regression, classification, object detection and semantic segmentation, but to understand this in-depth it is necessary to conduct a chronological analysis, from its origin to the most current versions.

One of the first networks of this type that appeared was AlexNet [62] – see Figure 2.9, a model based on CNN which was trained for the classification of images and that participated in the great challenge as ImageNet, specifically in the year 2012. Next, we find this proposal [63], which was able to convert a CNN network trained for classification like AlexNet, into a fully convolutional neural network (Fully Convolutional Neural Network (FCN)) that could be trained end-to-end and pixel-by-pixel with the task of improving the semantic segmentation [64].

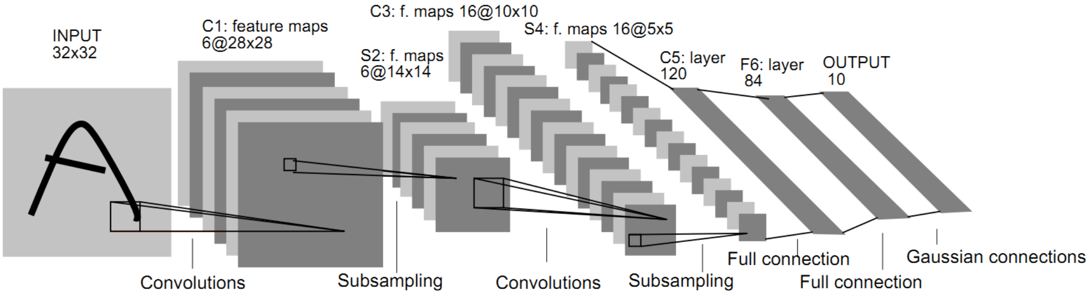


Figure 2.9: Architecture of AlexNet-5 [11].

This FCN model achieved state-of-the-art performance on datasets such as Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes (VOC), NYUDv2, Scale-Invariant Feature Transform (SIFT) Flow, among others, and has been of great relevance at a temporal level, since several works have adopted the CNN-based supervised learning approach to analyze remote sensing images, which directly relates to our proposal.

We then found a proposal [65] that was able to compare a set of 1D-CNN with convolutions in the spectral domain and a set of 2D-CNN with convolutions in the spatial domain to obtain a pixel-by-pixel class prediction, where his work concluded that the set of 2D-CNN was superior.

Taking the example of the existing literature and previous cases, other proposals [66] were based on the use of more efficient FCN-based models, such as SharpMask [67] or Multi-Path Refinement Networks for High-Resolution Semantic Segmentation (RefineNet) [68].

As we have already seen, CNN networks have evolved rapidly over the years, but to fully understand them and thus be able to present their possible variations, it is necessary to explain their internal operation in detail, since they are a type of neural network that has been specially designed to work with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data, where the central element of these is the convolutional layer that gives its name to the network and is where the operation called " convolution".

In the context of a convolutional neural network, convolution is a linear operation that involves multiplying a set of weights with the input, much like a traditional neural network, this is because the technique has been designed primarily for a two-dimensional input and therefore the multiplication is performed between an array of input data and a two-dimensional of weights, called a filter – see Figure 2.10 [69].

Within this processing, the filter must be smaller than the input data and the type of multiplication applied between a patch the size of an input filter, this being a scalar product, that is, the multiplication by elements between the patch of the filter size of the input and the filter, which is then added together, always resulting in a single value.

Specifically, filters are systematically applied to each overlapping part or filter-sized patch of the input data, from left to right and top to bottom, being this systematic application of the same filter on an image is a powerful idea since that being designed to detect a specific type of feature in the input, then systematically applying that filter to the entire input image allows the filter to discover that feature anywhere in the image [69].

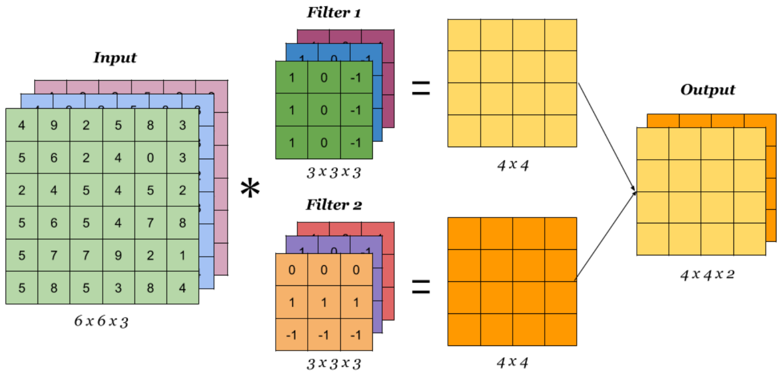


Figure 2.10: Operation of convolution with Multiple filters [12].

Once the basic operations have been explained, we are going to analyze their basic architecture by layers in-depth, taking as input an RGB image where the convolutional network will ingest the three separate layers of colour stacked one on top of the other and they will be considered channels.

Starting with the first layer of the CNN, we will find the Convolutional Layer, which constitutes the central building block and does most of the computational heavy lifting, whereas we have seen before, the data coming from the images is combined employing filters or kernels, applied through a sliding window, a process which involves taking the product of the filter elements in the image and then adding those specific values for each sliding action.

Continuing, the second layer is named, the Activation Layer, where an operation such as ReLU (Rectified Linear Unit) or a similar result was applied and thus a rectifying function was introduced to increase the non-linearity in the CNN network, a fact that benefits a correct output of the system, since as a general rule the images are made up of different elements that are not linear with each other.

In the third place, we will find the Pooling Layer, which has as its main objective a reduction in the sample of the characteristics and that is composed of hyper-parameters such as the dimension of the spatial extension or the Stride, that is, the number of characteristics that the sample skips. sliding window along the width and height.

Finally, we find the Fully Connected Layer, which in other words means that it involves a Flattening process, where the entire map matrix of grouped features is transformed into a single column which, in turn, is then fed back to the neural network for processing, with the layers fully connected, when we combine these features to create a model, to which we finally send an activation function, like a Softmax or a Sigmoid, to classify the output.

Some of the variations in the structures of Convolutional neural networks have made great advances in the field of artificial intelligence, of which a large sample may be the Imagenet contest, some of the most relevant versions being: VGGNet, GoogLeNet With Inception (different versions), Residual Neural Network (ResNet)... [70].

To continue, we will present other of the models derived from the classic architecture of the convolutional neural network (CNN), which is known as VGGNet and whose main development objective was to increase the depth of said CNNs to increase the performance – see Figure 2.11.

VGG stands for Visual Geometry Group and refers to a standard deep convolutional neural network (CNN) architecture with several layers, where the term ”deep” refers to the number of layers, thus finding VGG-16 or VGG- 19 consisting of 16 and 19 convolutional layers. The VGG architecture is one of the foundations of innovative object recognition models, mainly because being developed as a deep neural network exceeds baselines in many tasks and datasets beyond ImageNet.

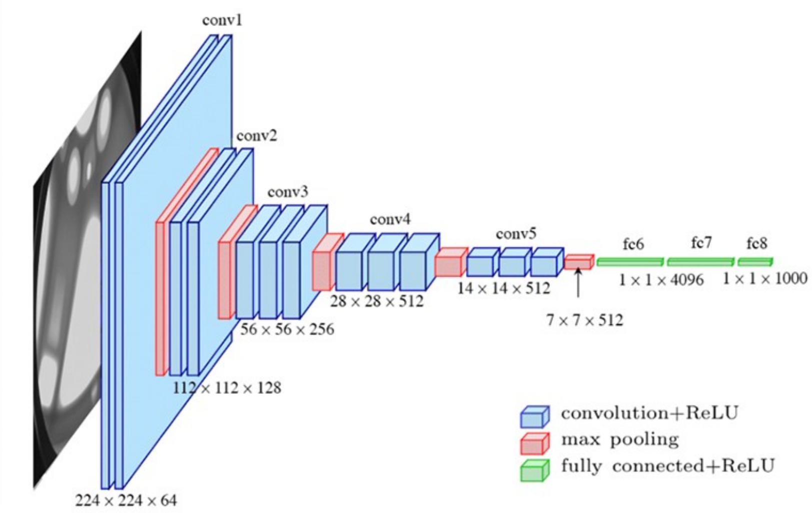


Figure 2.11: The architecture of VGG16 [13].

As mentioned, there are mainly two variants, the first VGG16 is an architecture that supports 16 layers, with 13 of these layers being convolutional and the other three layers fully connected. This proposal was submitted by A. Zisserman and K. Simonyan [71] and managed to achieve almost 92.7% accuracy in the top five tests on ImageNet .

Its basic modification from the more traditional structure is that it replaces the large kernel-size filters with several 3×3 kernel-size filters one after another, which makes significant improvements over AlexNet. Similarly, the VGG19 model has the same structure as VGG16 except that it supports 19 layers, this means that VGG19 has three more convolutional layers than VGG16.

As in anything, not everything could be advantageous and therefore one of the crucial disadvantages of the VGG16 network is that it is a giant network, which means that it takes more time to train its parameters and therefore, due to its depth and number of fully connected layers, the smallest VGG16 has more than 533 MB, being therefore the implementation of a VGG network a very expensive task [72].

Having highlighted the famous victory of AlexNet [73] in the Large Scale Visual Recognition Challenge (LSVRC) 2012 qualifying contest, surely the most innovative work in the machine vision/deep learning community in recent years was undoubtedly the Deep Residual Network [74], where a new model called ResNet, which makes it possible to train up to hundreds or even thousands of layers and still achieve convincing performance – see Figure 2.12.

Since the appearance of the ResNet model in 2015, which managed to impress people, many researchers have immersed themselves in its characteristics that concern its success and therefore, many improvements have been made in the architecture, thus allowing to take advantage of its powerful rendering capacity, and consequently boosting the performance of many machine vision applications other than image classification, such as object detection and facial recognition.

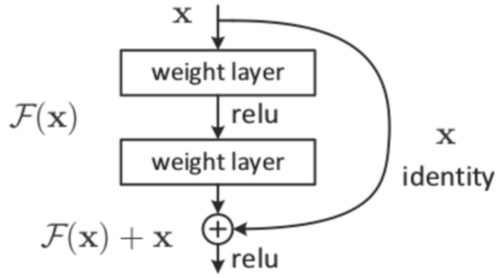

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.12: Typical layer in a Residual Neural Network [13].

We know that a feed-forward network with only one layer (according to the universal approximation theorem) is sufficient to represent any function, however, there is still the problem that the layer can be massive and therefore the network becomes prone to over-fit the data.

To solve this problem, a common trend has started to develop in the research community that network architecture needs to be deepened. Starting from the basis that the AlexNet architecture had only 5 convolutional layers, the latest generation CNN architecture goes deeper and deeper, reaching cases like the VGG network [75] and GoogleNet (Inception-v1) [76] that had 19 and 22 layers respectively.

Although this may seem like a solution, the reality is that deep networks are difficult to train due to the notorious disappearing gradient problem, where this gradient propagates back to previous layers and therefore repeated multiplication can cause the gradient to be infinitesimally small, with the result that as the network gets deeper, its performance saturates or even begins to degrade rapidly.

This problem is known as ”the leakage gradient problem” and there are several ways to solve it, one of them [76], adds an auxiliary loss in an intermediate layer as additional supervision, although none manages to fully address the problem and they are just little tweaks.

One of the most promising proposals [74], argued that the process of stacking layers should not degrade network performance, mainly because identity mappings could be stacked on the current network, and therefore the architecture The resulting model would have the same performance, implying that the deeper model should not produce a larger training error than its shallower counterparts.

The previously detailed process is called ”identity direct access connection” which is based on the omission of one or more layers and was not implemented in a pioneering way by the ResNet model, but Highway Network [77] was responsible to introduce closed direct access connections. This process works similarly to gates, which are parameterized and therefore control the amount of information that is allowed to flow through the shortcut – see Figure 2.13.

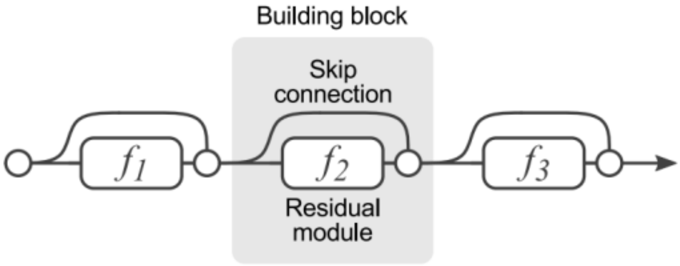


Figure 2.13: Example of ’Unravelling’ a Residual Neural Network [13].

A similar idea can also be found in one of the previously explained models, specifically the Long Term Short Memory (LSTM) [78], in which there is a parameterized forgetting gate that controls how much information will flow to the next time step. Although this seemed like an adequate solution, the results did not turn out to be as significant as expected and it is better to keep these ”slope roads” clear than to opt for a larger solution space.

Therefore, and following this idea, a new model was proposed where the residual block was refined and a pre-activation variant of the residual block was attached [79], in which gradients can flow through direct access connections to another previous layer without obstacles, obtaining great results, which served to quickly position this ResNet model as one of the popular architectures in computer vision.

Another the variant of the CNN structure was proposed by Huang [80], where he imagined a novel architecture called DenseNet that was in charge of further exploiting the effects of direct access connections, magnifying the idea and trying to connect all the layers directly between each other.

Working in such a way that in this new architecture, the input of each layer will consist of the feature maps of all the previous layers and its output is passed to each subsequent layer, these feature maps being the ones that would be added with depth concatenation. With this new structure, [81], not only was it able to address the problem of vanishing gradients raised earlier, but it was also able to encourage feature reuse, which makes the network highly efficient in terms of the number of features parameters – see Figure 2.14.

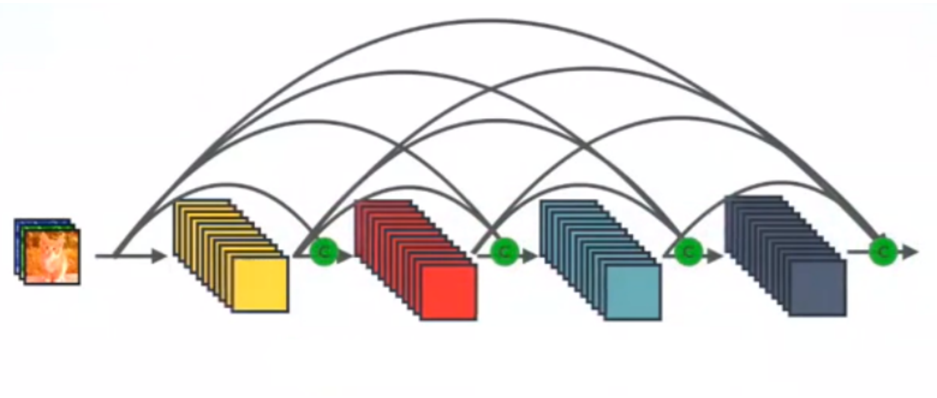


Figure 2.14: One Dense Block in DenseNet [14].

For a simple form implementation, one has to get used to the idea that the identity map was added to the next block, which could impede the flow of information if the two-layer feature maps have very large distributions. Consequently, the concatenation of feature maps is capable of conserving them all and, in turn, allows increasing the variance of the results and thus promoting the reuse of features.

Another noteworthy factor was the problem that the network grew too large, for which they used a 1x1 convolutional bottleneck layer to reduce the number of feature maps before the expensive 3x3 convolution, as well as implemented a hyper-parameter called growth rate, to control these values.

As the last variant of the basic CNN architecture, you find the Inception network, which was not only an important milestone in the development of CNN classifiers but also broke with the old idea of where to improve the model they stacked deeper and deeper convolution layers, to get better performance.

This network was designed in a complex and resistant way and used a variety of tricks to boost performance, both in terms of speed and accuracy, which led to constant evolution and the creation of several versions of the same, which we will deal with in-depth below. Each of these versions iteratively improved the previous version and the fact of being able to understand the improvements included in these new versions will help us create custom classifiers optimized for both speed and accuracy.

Starting then with Inception-v1 [76], we are faced with the problem of locating the same element in a wide variety of images, but for each of these, the element changes location, shape and size, making it difficult to choose the size of adequate nucleus for the convolution operation in each of the cases, because if a very large nucleus is chosen, the information will be distributed more globally or generally, however, if a smaller nucleus is preferred, the information will be distributed more locally.

Due to this problem, taking into account that very deep networks are prone to over-fitting and that we still maintain the difficulty of passing gradient updates through the entire network, a naive stacking of large convolution operations will occur, which as we already know, is computationally very expensive.

To solve this, the idea of having multiple filters, with multiple sizes operating at the same level, is proposed, thus creating a much ”wider” network, instead of a much ”deeper”. This was named the ”Inception” module, after which the model is also named, and it is mainly based on one input, with 3 different sizes of filters (1x1, 3x3, 5x5), for which the maximum grouping is done where the outputs are concatenated and sent to the next startup module – see Figure 2.15.

As has been indicated throughout the entire section, deep neural networks are computationally expensive, and this also applies to the module created, in this way and to make it more economical, the authors limited the number of input channels by adding an additional convolution of 1x1 before the convolutions of 3x3 and 5x5.

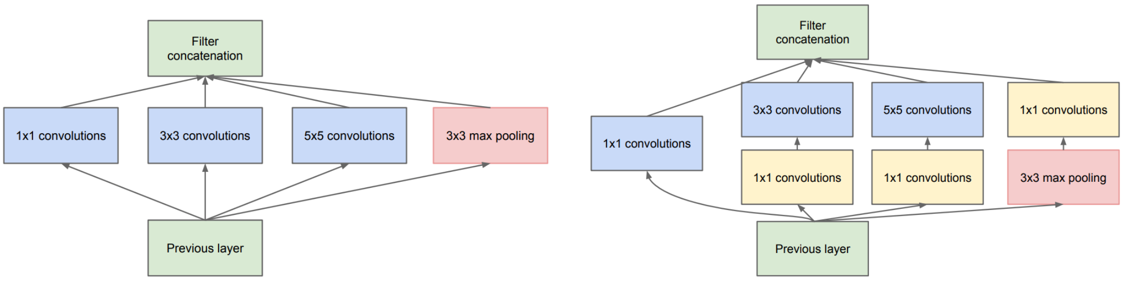


Figure 2.15: Example of Inception-v1 Module [14].

Using the explained module, exactly using the one that contained the reduced dimensions, a neural network architecture was formed, which would be popularly known as GoogLeNet (Inception v1), containing a total of 9 initial modules of this type stacked linearly, 27 depth layers of which 5 are pooling layers and adding the global average pooling to the end of the last starting module.

Despite having created this new module that helped with the problem of depth, the number of parameters and that facilitated its implementation, as with any very deep network, it was still subject to the problem of the gradient of disappearance. To prevent the gradient from becoming extinct as it passes through the model, the authors had the idea of introducing two auxiliary classifiers, where a Softmax function was applied to the outputs of two of the initial modules and an auxiliary loss was calculated on the same labels.

Following in Inception 2 [82], the authors proposed a series of updates that not only increased the accuracy of the model but also reduced the computational complexity. To do this, he will attack the problem called ”bottleneck”, starting from the idea that neural networks work better when the convolutions do not drastically alter the dimensions of the input, in this way, they based their efforts on avoiding the drastic reduction of the dimensions which ends up directly causing the loss of information.

In this way, the 5x5 convolution was factored into two 3x3 convolution operations, to improve the calculation speed, since a 5x5 convolution is 2.78 times more expensive than a 3x3 convolution and when creating two 3x3 convolutions indeed leads to an increase in performance. In addition to this modification, convolutions of filter size nxn are also factored into a combination of 1xn and nx convolutions, leading to a reduction of 33% compared to the single 3x3 convolution – see Figure 2.16.

With this, they managed to increase the number of filter banks in the module, that is, they became wider instead of deeper, and in turn, the rendering bottleneck was eliminated.

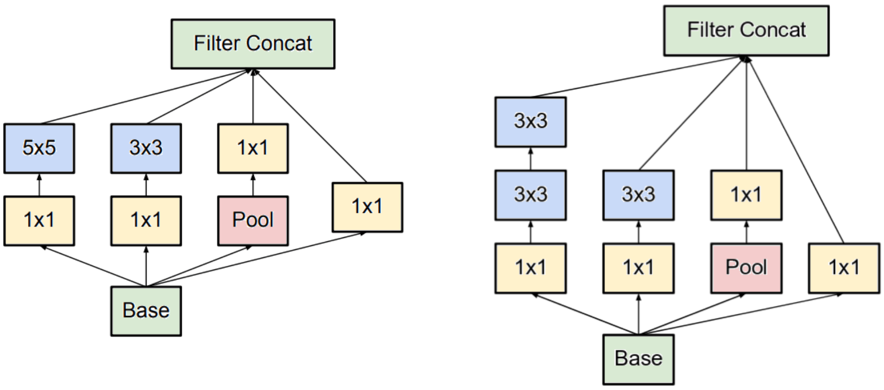


Figure 2.16: Example of Inception-v2 Module [14].

Continuing with Inception-v3, the authors noticed that the helper classifiers did not contribute much until near the end of the training process and this only happened when the precisions approached saturation and that's when the idea to improve Inception v2 arose without drastically changing the internal composition of the modules [83].

In this way, Inception-v3 implemented all the previous updates indicated for Inception v2, but also included, an optimize Root Mean Square Propogation (RMSProp), factored 7x7 convolutions in the same way that the 5x5 had been factored in v2, BatchNorm in the Auxiliary Classifiers and finally a Label smoothing, which is a component that is responsible for regularization and avoids overconfidence or excessive adjustment of the system – see Figure 2.17.

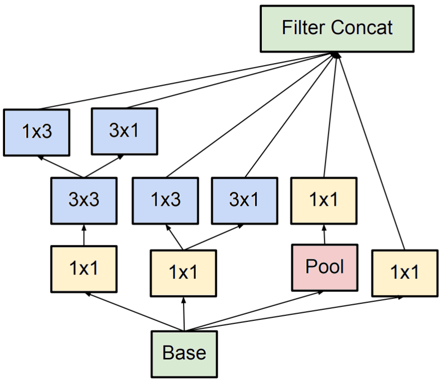


Figure 2.17: Example of Inception-v3 Module [14].

We finally got to Inception v4 [84], where the authors realized that some of the modules were too different, while other modules had become more complicated than necessary, and therefore made the decision to standardize the modules to increase performance.

For this reason, in this case, the blocks that make up the seventh were not modified, but changes were proposed in the ”stem”, that is, the initial set of operations executed before introducing the Inception blocks, as well as the ”Inception Blocks”. special ”shrink” that were used to change the width and height of the grid.

Inspired by the performance of ResNet, a hybrid module was proposed, with the premise of introducing residual connections that add the output of the start module's convolution operation to the input. For this new module, there are two subversions of Inception ResNet v1 and v2, where the Inception-ResNet v1 version has an estimated computational cost similar to that of Inception v3, while the cost of Inception-ResNet v2 is more similar to that of Inception v4 – see Figure 2.18. [85]

To achieve this goal and therefore for the residual sum to work, the input and output after the convolution must have the same dimensions, otherwise, it would not be mathematically possible, therefore they were forced to use 1x1 convolutions after the original convolutions, to equalize depth sizes, since the depth increases after the convolution, while the grouping operation inside the main bootstrap modules was replaced in favour of residual connections.

The parts of the network with residual units that were in deeper points of the architecture, caused the network to shut down if the filter number was raised too high, of the order of 1000, to solve it and therefore provide stability to the network, residual activations were scaled by a value of about 0.1 to 0.3, thus achieving Inception-ResNet models that could achieve higher accuracies at a lower epoch [86].

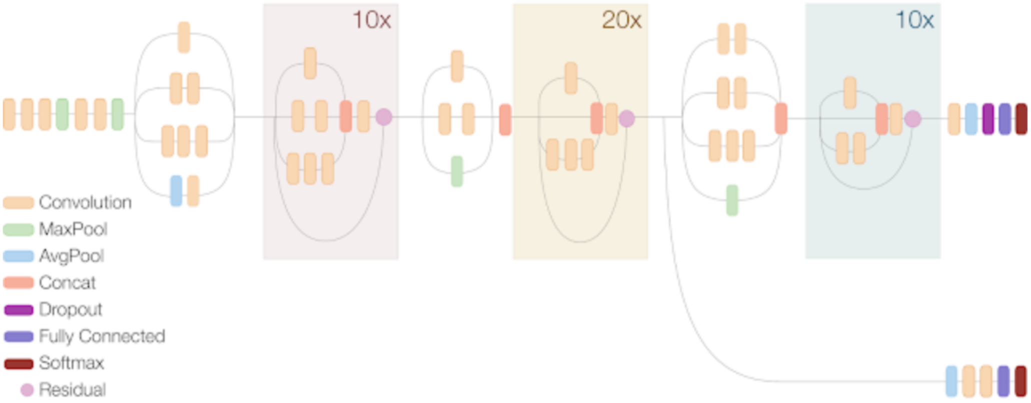


Figure 2.18: Schematic diagram of Inception-ResNet-v2 [14].

2.1.3 The Transformer Model

Finally, once the common structures have been explained, the structure on which our project is based is going to be explained, this being one of the most innovative and complex that exists in the world of AI.

This structure called, Transformer Neural Network, is a novel architecture and was proposed in the document ”Attention is all you need” 2017 [87], whose main objective is to solve tasks sequence by sequence while handling long-range dependencies with ease, thus becoming one of the most advanced techniques in the field of Natural Language processing.

A quick example of its capacity to act, we obtain it when comparing it with a Recurrent Neural Network (RNN), since they are the other type of networks that are capable of retaining information and have dependencies, but unlike this one that needs to send the sequence of data from Continuous and dosed form, the structure of Transformer allows the entire dataset to be sent at the same time, which not only facilitates the task but also allows the system to have access to extra information, such as the position and location of each data.

Below is the Figure 2.19 of the structure of the Transformer, where it can be seen exactly all the parts that compose it, which are explained below and provides an overview of its internal structure.

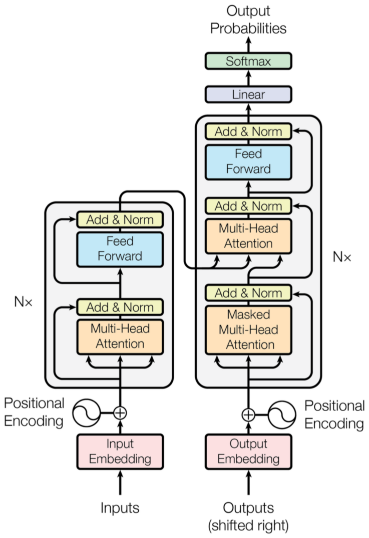


Figure 2.19: The Transformer Model Architecture [15].

The internal structure is mainly divided into two large blocks and these in turn into smaller blocks that constitute them. The large blocks would be the encoder and the decoder, the first being the one formed by the left half of the diagram and the second by the right half (as shown in the image and indicated in the annotations) – see Figure 2.20.

As can be seen, the encoder block is mainly made up of data input, a Positional encoding block, a Multi-Head Attention block and a Feed-Forward block. The operation of this block is quite basic, although it includes elements that require a detailed explanation to facilitate its compression. The data would enter the system from the main entrance, where we would find ourselves in the Embedding Space, an open space where the data with similarities are grouped or rather would be present close to each other in that space, thus converting our input data into a vector.

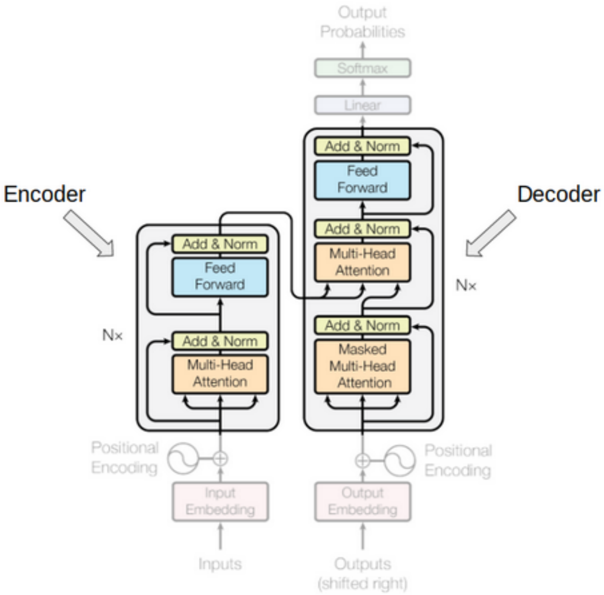


Figure 2.20: The Transformer Blocks [16].

This vector will be sent to the Multihead Attention block, where each data is analyzed according to how relevant it is compared to the data that are close to it and will be represented as an attention vector, where for each data, we can have a generated attention vector, which captures the contextual relationship between the data in that space. Although the reality is that a single attention vector is not obtained for each data/element, since the attention will vary according to the different approaches and therefore the solution lies in determining multiple attention vectors per data and taking a weighted average to calculate the final care vector of each one.

Finally, we must be able to convert these attention vectors to a different format which is acceptable for the next encoder or decoder layer and this is where the Feed Forward Network accepts attention vectors (one by one, but already with the dependencies contextual data) and sends them, at the same time, to the coding block, to obtain the set of vectors encoded for each data simultaneously.

Once the explanation of the encoder block is finished, we will start with the decoder, whose input is similar to the previous block, where the embedding layer and the positional encoder part, change the input data and transform it into the respective vectors.

Similarly, this data vector is sent to the self-attention block, where attention vectors are generated for each data according to how much each data is related to the elements that accompany it and give it context, but unlike the previous case, this block is “Masked” since our intention is for the system to learn to recognize the patterns and therefore, part of the input data must be hidden so that the system is forced to predict the output.

Once the prediction process is finished, the attention vectors resulting from the previous layer and the coding block vectors (previously obtained and connected as shown in the diagram) are sent to another multi-head attention block and this is in charge of establishing the relationships that will exist between the encoder and decoder data.

To continue, the output data will be sent to the Feed Forward Network and as in the encoder block it will cause the output vectors to be formed into something that is easily acceptable by another decoder block or a linear layer, as is the case, then which again fulfils the feedback function, so that the dimensions are expanded in data numbers.

Finally, the data will be sent to a Softmax layer, which transforms the input into a probability distribution, which is interpretable by humans and which will produce a final output with the element/data with the highest probability [88].

Once the basic operation of the Transformer architecture has been explained, this second part of the section explains the specific case of the Vision Transformer (ViT) model, which is a model that was presented in a research article published as a conference paper in International Conference on Learning Representations (ICLR) 2021 [89] and which was developed and published by Neil Houlsby, Alexey Dosovitskiy, and other authors who were part of the Google Research Brain Team.

The ViT model is a visual model based on a Transformer architecture which was originally designed to perform text-based tasks, however, the ViT architecture represents an input image as a series of image patches, similarly to the series of word embeddings used when working with text Transformer models, and in this way, it can directly predict the class labels for the image. It should also be noted that the self-attention layer in the model makes it possible to embed global scope information in the general image, allowing the model to also learn training data to encode the relative location of image patches to reconstruct the image structure – see Figure 2.21 [90].

In this way, we find that this type of Transformers obtains high success rates when it comes to Natural Language Processing (NLP) models and that they are also applied to images for image recognition tasks. But while networks like CNNs use pixel arrays, the ViT model splits the images into visual tokens of fixed size, which correctly embeds each of them and includes the positional embedding as input to the Transformer encoder.

With these features, ViT models achieve remarkable results compared to convolutional neural networks while requiring less computational resources for pre-training, exhibiting extraordinary performance when trained on sufficient data, and outperforming a similar state-of-the-art CNN with 4 times less computational resources [91].

In general, the Transformer encoder usually includes three types of layers in its architecture, the Multi-Head Self-Service Layer (Multi-Head Self-Service Layer (MSP)), which is responsible for concatenating all the service outputs linearly in the correct dimensions, being the large number of attention heads, which help to learn local and global dependencies in an image. Next, we find the Multilayer Perceptron Layer (MLP), which contains a two-layer Gaussian Error Linear Unit (Gaussian Error Linear Unit (GELU)) operation, and finally, the Layer Norm (Layer Norm (LN)), which is always included at the end of the layers and mainly serves to improve training time and overall performance since it does not include new dependencies [92].

In addition, both residual connections and the self-service mechanism are included. The former is placed after each block and allows information from other layers of the model to flow through the network directly without going through non-linear activations, while self-attention is one of the essential blocks of machine learning Transformers and that It allows quantifying the interactions of entities by pairs that help a network to learn the hierarchies and alignments present within the input data, making vision networks more robust and thus proving to be a key element.

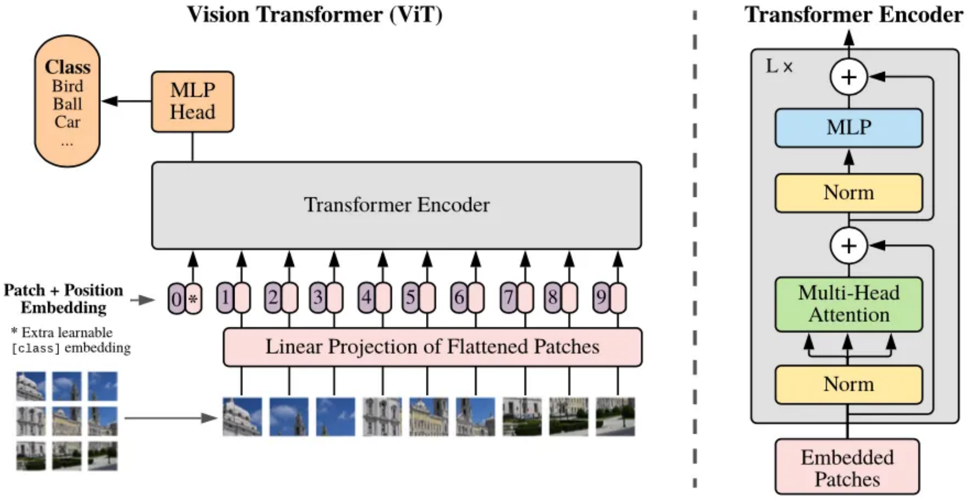


Figure 2.21: Vision Transformer ViT Architecture [17].

In the same way that they work in the rest of the models explained above, the performance of a vision Transformer depends on the depth of the network, the specific hyperparameters of the dataset and decisions such as those of the optimizer, this last point is somewhat relevant, since ViTs tend to be more difficult to optimize compared to CNNs.

As already mentioned, ViT models must be trained with large datasets, being a realistic case, that a model trained with a database containing more than 14 million images, can outperform CNNs with no problem, however, if you don't have a dataset of these dimensions, your best bet is to stick with ResNet or Improving Accuracy and Efficiency through AutoML and Model Scaling (EfficientNet) [93].

As we have seen from their good functionality, it is not surprising that vision Transformers tend to excel in popular tasks such as image segmentation and classification, action recognition, object detection or image recognition. As well as in generative modelling applications and multi-model tasks, some of the most relevant being visual grounding, answering visual questions and visual reasoning.

Other real applications in which this type of model is already being included are parts of video processing, such as video forecasting and activity recognition, in addition to image enhancement, colourization and super-resolution, which is also proposed for tasks that include 3D analysis, such as point cloud segmentation and classification.

2.2 Related Work

In this second part of the chapter and once the fundamental bases that allow a more detailed explanation has already been established, the issue or task that we are trying to solve through this model proposal is addressed, the Image Captioning, specifically in the field of Remote Sensing Image Captioning.

As the section progresses, it will be demonstrated that although Remote Sensing Image Captioning is a derivative of Image Captioning, it contains some unique characteristics that greatly differentiate it and that make it so easy to work with it. At the same time, topics such as the use of information extracted from images, the possible integration of metadata such as geographic information and the possible and future applications in which this technology will have a fundamental role are also explored.

2.2.1 General Image Captioning

The image captioning is responsible for generating a semantic description of a given image, which will be more or less detailed depending on the application for which they are required. This sounds great, however, it is still a big challenge to close the “semantic gap” between low-level features and high-level semantics in remote sensing images, mainly due to their difficulty in interpretation – see Figure 2.22.

Despite this, state-of-the-art examples of this problem have recently been achieved through deep learning methods and thus deep learning models are capable of achieving optimal results in the field of caption generation problems better than other similar models. It is also important to note that using this type of model does not require complex data preparation or a series of specifically designed models, a single end-to-end model can be defined to predict a title, given a photo [94].

The image captioning task for the generation of a description has therefore been one of the most relevant and fundamental tasks in the Deep Learning domain, due to its large number of applications and its versatility in different media, economic, artistic, medical... A great example of this versatility comes from the hand of the NVIDIA company which is making use of these image captioning technologies to create an application to help people who have little or no view to be able to detect emissions from different sources, something similar to help through sign language for deaf people [95].

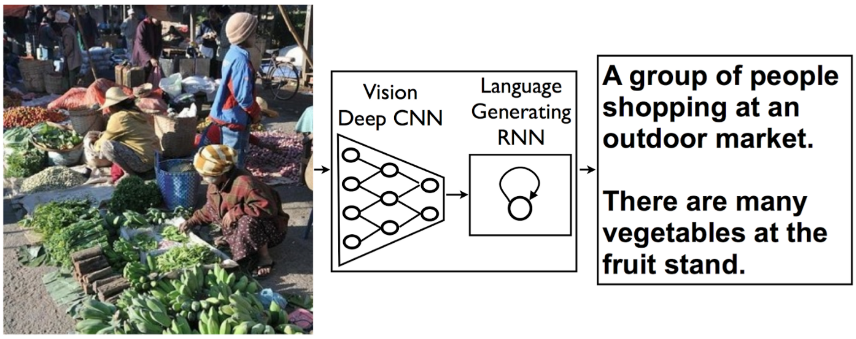

Vision
Deep CNN
Language
Generating
RNN
A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Figure 2.22: Image Captioning Basic Scheme [18].

Image captioning methods have been around since the advent of the Internet and its common use as the primary medium for sharing and sending images. One of the first variations, proposed by Krizhevsky [96], implemented a neural network using unsaturated neurons and a very efficient single-method GPU implementation of the convolution function. The proposed net is made up of eight layers with weights, the first five are convolutional and the remaining three are fully connected, and the output is fed to a softmax which produces a distribution over the 1000 class labels. This proposal maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average across training cases of the log probability of the correct label under the prediction distribution. Also, it is applied a method for regularization that will largely reduce over-fitting called ”dropout”, which consists of setting to zero the output of each hidden neuron with a fixed probability, being the neurons which are “dropped out” not contributing to the forward pass and not participating in backpropagation.

Deng [97] was able to introduce a new database (for that moment), which we currently know very well, since it is one of the most used called ImageNet [98], which constitutes an extensive collection of images created using the core of the WordNet structure (currently just the nouns) in which each node in the hierarchy is represented by hundreds and thousands of images and which has been instrumental in advancing computer vision and research into deep learning, where all data is freely available.

Karpathy and FeiFei [99] used datasets of images and their specific descriptions, constituted in a set of sentences, to learn about the inter-modal correspondences and the complex interrelation that existed and exists between visual data and language, all through a multi-modal recurrent neural network that uses the inferred arrangement of features. The alignment model proposed learned to ground dependency tree relationships with image regions for a classification purpose, using a bidirectional recurrent neural network to compute word representations, with no need to compute dependency trees and allowing unlimited interactions of words and their context. At the same time to establish the inter-modal relationships, the authors represent the words in the same dimensional embedding space that the regions of the image occupy, but instead of projecting each word directly onto this embed, the idea was to use a bidirectional recurrent neural network (BRNN) to compute the dependency tree relationships, imposing also an arbitrary maximum size of the context window and the use of dependency tree parsers.

Yang [100] proposed a system for the automatic generation of a description, in natural language, of an image, based on a multi-model neural network method closely related to the human visual system that automatically learns to describe the content of images. The model consists of two sub-models: an object detection component, which extracts the information of objects and a localization model, which extracts their spatial relationship in images, so each word of the description will be automatically aligned to different objects of the input image when it is generated. This approach would significantly help improve the visual understanding of the image through a multi-model neural network, composed of specific modules for the detection and localization of objects and that tried to artificially replicate how the human visual system can learn to describe the content in which it is found through images automatically.

Despite the great advances in image captioning, using Recurrent Neural Networks powered by long-short-term-memory (LSTM) units, mitigating the vanishing gradient problem, and the ability to memorize dependencies, LSTM units are complex and inherently sequential across time. Addressing this problem, Aneja et al. [101] proposed a new convolutional image captioning technique based on recent works about the benefits of using convolutional networks for machine translation and conditional image generation. The model is based on the convolutional machine translation model and contains three main components, two word embeddings and masked convolutions, being this last one a feed-forward without any recurrent function, and since there are no recurrent connections, all ground-truth words are available at any given time-step so the model can be trained in parallel for all words.

Some new architectures were studied exhaustively by Pan et al. [102] where, through a set of network architectures containing large datasets with different content styles, wanted to discover correlations between image features and keywords, and automatically find good keywords for a new image, proposing a total of 4 methods (Corr, Cos, SvdCorr, SvdCos) to estimate a correlation-based translation table and being the experiments performed on 10 different image datasets. The study also includes the idea of ”blob-tokens”, labels of assigned image regions (represented by a vector of features regarding its colour, texture, shape, size and position) closest to a cluster centre. Being the accuracy of image captioning affected directly by the quality of these blob-tokens, generated by using the K-means algorithm on feature vectors of all the image regions.

In turn, Vinyals et al. [103] using the deep recurrent architecture, was able to present a generative model that took advantage of recent advances in both machine translation and computer vision technology, combining and using them to generate natural and fluid descriptions of an image of in such a way as to ensure a higher probability that the generated sentence accurately describes the target image and its context. The proposed approach presented an end-to-end neural network system that can automatically generate a reasonable caption by an input image, based on a convolution neural network that encodes an image into a compact representation, followed by a recurrent neural network that generates a corresponding caption. Being the model trained with the objective of maximising the likelihood of the generated caption given the input image.

It is worth mentioning Xu et al. [104] who finally introduced a model based on the attention mechanism which was able to learn to describe the regions of the image automatically, specifically focusing on the different points of the image as the text sequence progressed. The study introduced two attention-based image caption generators under a common framework, a “soft” deterministic attention mechanism and a “hard” stochastic attention mechanism, being the first one trainable by standard back-propagation methods, while the second one by maximizing an approximate variational lower bound, and in this way, allowing both to be able to identify object boundaries and, at the same time, generate an accurate descriptive sentence.

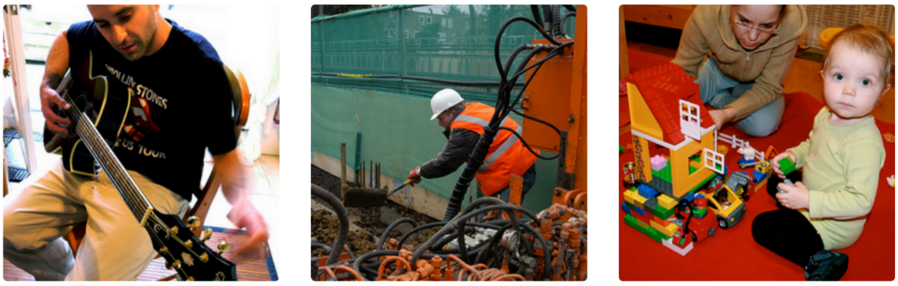
Despite these great advances in the subject, the image captioning task can still be considered as an end-to-end sequence-to-sequence problem, since it converts images, which are considered a sequence/string of pixels, into a sequence/ chain of words and therefore, for this, we need to process both the language section, which would correspond to the descriptions created and the visual section, that is, the captured images [105].

Due to these remarkable characteristics, image Captioning has become one of the great current issues within artificial intelligence and has made great progress in recent years, making it more interesting by combining both Computer Vision and Natural Language Processing – see Figure 2.23 [106].

To give some context to all this, it should be noted that similar to many technological advances, in the beginning, it was considered impossible for a computer to describe an image, but as deep learning techniques were improved and the storage and processing of the large volumes of data available, the possibility of creating complex models whose main task is to generate legends that describe an image was considered.

This breakthrough in image captioning is mainly due to the rapid and efficient development of deep learning, a sub-field of machine learning related to algorithms and inspired by the structure and function of the brain, considered by the scientific community as one of the fastest advancing fields of study and research, which is making its way into all of our daily lives without even realizing it [107].

The technologies needed to convert the sequence of pixels, which represents an image, into a sequence of characters, which would correspond to a text or a description, began their drastic advance about five years ago. As time went by, better performance, precision and reliability were achieved, which makes it possible for this task to generate much more fluid and efficient descriptions for the areas in which it needs to be applied, which can range from social networks to commerce. electronic, even becoming a technological solution to help blind people discover the world around them [108].



"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

Figure 2.23: Example of Image Captioning [19].

When analyzing the different methods that exist for the image captioning task, we can find 3 large groups: template-based, recovery-based and encoder-decoder based image captioning.

Template-based methods generate sentences by filling in the blanks with various goals, attributes, and behaviours extracted from the image. In this way, the sentences that are generated using this method are grammatically correct but inflexible and stereotyped due to their predefined templates and therefore do not have the versatility to adapt to various applications, since they only follow a pattern.

On the other hand, retrieval-based approaches create a database that includes pairs of images and their corresponding text descriptions. Thus, for a given image, a search for similar images and their corresponding descriptions in the database is performed, and then the sentences that correspond to the similar image found in the database are chosen as the caption of the query image. In this way, the sentences generated by this type of model are correct in syntax and add diversity to the structure of the sentence, as long as the database has been built correctly.

The encoder-decoder methods, which are the most used in the field, since they are normally characterized by using a CNN as an encoder to extract features from the image introduced into the system and incorporate an Recurrent Neural Network (RNN) or LSTM as a decoder to generate sentences with those characteristics of the image. Even though this is usually the most used structure, as shown by recent studies presented in this section, different structures differ from the classic ones and that can provide better results. The most relevant for our work is the Transformer-based encoder-decoder model.

Although, as we have seen, there are different methods to complete the task, we will focus mainly on the last one, which, in general terms, is normally composed of three main components: Image encoder, Text decoder and a Caption generator.

In this way, the logical operation would have the following scheme, first, the Image encoder takes the source photo as input and produces an encoded representation of it that captures its essential characteristics, that is, it progressively extracts the different simple geometric shapes as curves and half circles in the initial layers and progresses to higher-level structures such as noses, eyes and hands, finally identifying elements such as faces or wheels.

Next, the text decoder takes the coded representation of the photo, which has been extracted by the Image encoder and generates a sequence of cards that describes the photo. This process is performed cyclically, generating a prediction in a loop, creating a token at a time which is then fed back to the network as input for the next iteration. Finally, the Caption generator takes the sequence of tokens extracted by the Text decoder and generates a caption that is a sentence of words in the desired language and that corresponds to a description of the input photo [109].

Before continuing, it is important to explain that the encoder-decoder method is not usually used in a basic way, but rather, as we will see throughout the work, it can be modified to improve its efficiency and results, the two most important being: encoder-decoder with attention and with Transformers – see Figure 2.24.

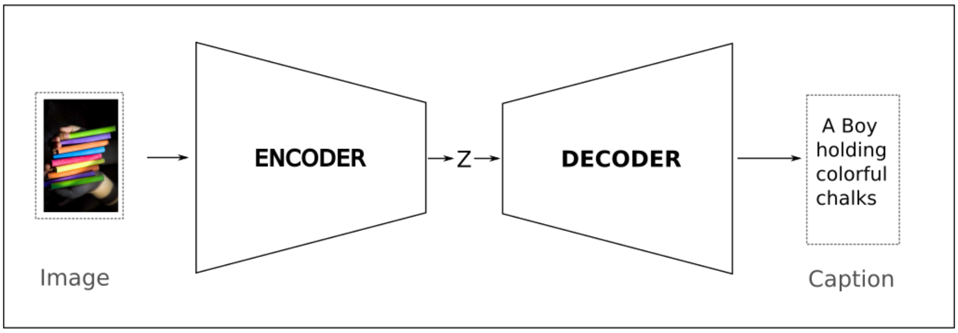
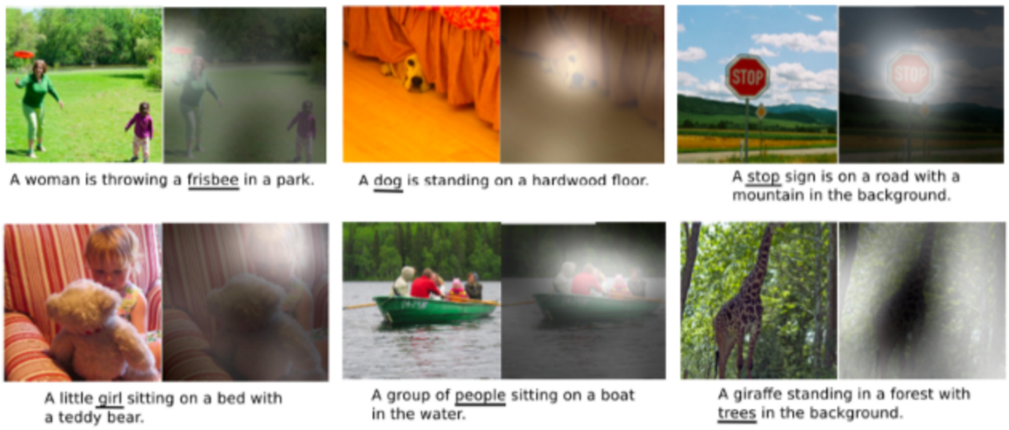


Figure 2.24: Encoder-decoder model for image Captioning [20].

The first, namely the attention encoder-decoder, is based on the premise that in recent years, the use of the attention mechanism in NLP tasks has greatly increased, as it has been found to significantly improve the performance of applications that use it, because as the model generates each word of the output, Attention helps it focus on the words in the input stream that are most relevant to that output word.

Therefore, it is normal that the attention mechanism has also been used together with this architecture so that as the Text decoder is producing the sequence of tokens, the attention mechanism acts to help it to concentrate on the part of the image that will be considered most relevant at any given time. In other words, the attention module attaches an image vector that is encoded together with the output token produced by the decoder, this vector contains a weighted attention score, which is combined with the image and increases or decreases the weight/ the relevance of those pixels that the model should focus on when predicting the next token.

An example could be having a picture with a cat, hiding under a blanket/sheet. When generating the caption corresponding to the input image, the approach of the model will vary, depending mainly on the token it is referring to at the moment. If we consider that the output phrase is ”An orange cat under a blanket”, the words that refer to the cat will cause the attention mechanism to focus on them, while those that refer to the blanket will focus on the cat – see Figure 2.25.



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Figure 2.25: Example of Attention Mechanism [21].

On the other hand, we find the Transformer mechanism, whose structure has already been discussed earlier in the work and which is mainly characterized by having an attention mechanism at its core without using an RNN structure.

For this mechanism, a few different variants have been proposed to address the problem of image captioning, these mainly try to solve the problem of simple encoding, in such a way that not only the individual objects are analyzed in the image, but also its spatial relationships within it are taken into account, since this is very relevant to fully understand the scene and to be able to make a much more consistent description, thus allowing us to know if an object is below, behind or to the side. side of another object and thus providing useful context and extra information for generating the description and context of the action.

To conclude this section, it is important to highlight the great importance and relevance that Image Captioning technology has acquired, mainly due to its versatility and the wide range of areas it affects, some of the most relevant areas today, which we will explain below.

One of the most relevant would be the labelling of E-commerce and online catalogues, where Artificial Intelligence is allowing the automatic creation of labels by photo, which can directly simplify the lives of users when they load an image in an online catalogue, whether for clothing or any other product because the AI will recognize the user’s image and generate attributes, such as categories or descriptions (mainly depending on the product or service sought) and thus determine the characteristics of the item that is you need either material, colour, pattern or dimensions [110].

On the other hand, this service in combination with an option to share stored photos to the online catalogue website for the creation of an automatic meaningful description of the image can mean a great advance in Search Engine Optimization (SEO) and organization tasks for the company itself, as well as a means of control for the authorities and regulatory bodies, so that it is verified if the image complies with the rules of the platform where it is going to be published, meaning greater control and increase in information traffic and therefore as much of the income.

Another great advance in the medical and accessibility area for daily and independent life would be the descriptions of automatic images for blind people, where the complete transformation of the environment captured through images would be needed, in a precise description in text and a modulator of voice capable of transmitting this information to the subject.

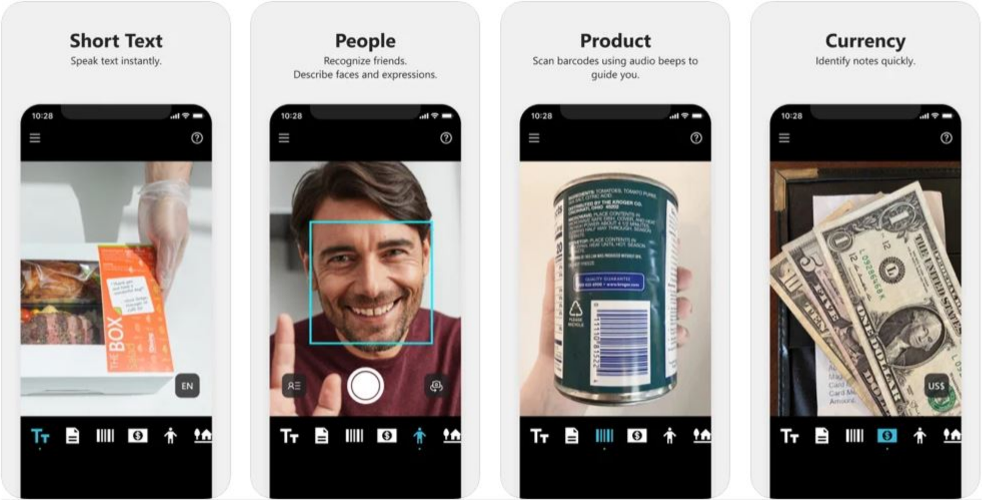


Figure 2.26: Microsoft Seeing AI app [21].

Although there are already some similar applications such as ”Seeing AI”, an application developed by Microsoft allows people with eye problems to see the world around them using smartphones, where the program can read both printed and handwritten text when the camera is pointed at him and gives audible indications after having identified objects and people. The main problem lies in the fact that this system is not yet integrated into an accessible and useful element, since although its implementation in a smartphone is ingenious, the fact of having to point where you want to walk makes it difficult to implement it in real life in society – see Figure 2.26 [111].

Another significant application, and more so in current times, where direct contact has been drastically reduced and the greatest number of relationships and interactions are implemented through social networks, is the creation of descriptions for them. Where an automatic caption could be generated through the tools provided by AI, being also an advantage, since the model would become increasingly intelligent because it would receive new input images almost continuously and that would allow it to recognize new objects, actions and patterns.

Although this system already exists and is currently implemented by both Facebook and Instagram, it is a great example of how this new technology is capable of adapting to our daily lives without us even realizing it [112].

Finally, a somewhat novel example is DeepLogo, a neural network based on the TensorFlow object detection API, which allows logos to be recognized and once identified, sets the name of the company or company to which it refers as the title, being able to catalogue messages, reports or studies automatically without the need for human intervention and therefore facilitating subsequent processing [113].

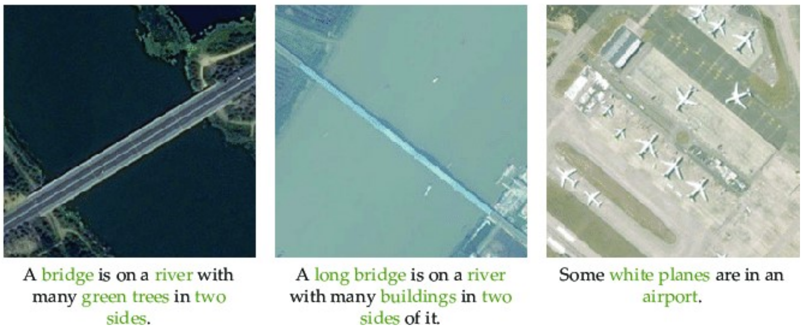
As we have already seen, image captioning has a wide variety of applications and even though only some of the most relevant ones have been explained, others can range from editing applications, use in virtual assistants, image indexing... covering fields as diverse as, advertising, medicine, security or even finance.

2.2.2 Remote Sensing Image Captioning

Remote sensing imagery captioning is a specialized application constrained by the type of images used, this is mainly because the output of the system can be used to describe in the form of well-developed sentences and to provide sharp detail over a large number of domains, such as the evaluation of the quality of the land for cultivation, the analysis of a crowd, the detection of its types and the classification of these crowds, the detection of traffic according to the main characteristics of the vehicle, such as model, colour or shape, the type of activity that takes place in the particular area, among others.

Image captions, as we have seen in the previous section, have always been considered as a combination of image processing, which would be Computer Vision and a text-processing model that would correspond to Natural Language Processing, thus being the most commonly recognized complication. Image captioning is the process of combining the worlds of image and text, using a compatible dataset to process through a computer – see Figure 2.27.

These databases for captioning remote sensing images often contain individual objects present in an easily identifiable manner, the images of which are often captured by drones which resemble near real-time, causing a user-friendly dataset to be created. and that, in turn, contains a diverse variety of objects, thus allowing that when training the model, it must become more efficient to be able to find the correct object and, in turn, relate it to the word indicated that allows summarizing the image in a concise, clear and above all-natural way.



A bridge is on a river with many green trees in two sides.

A long bridge is on a river with many buildings in two sides of it.

Some white planes are in an airport.

Figure 2.27: Visualization of remote sensing image captioning examples [22].

Similar to the field of captions for normal images, in the case of remote sensing images, the problem involves word processing to generate captions, which has evolved into a mathematical problem, as the image and text are converted to numbers to be processed by models [114].

Image captioning is a task through which a natural semantic description is generated from a given image, and this plays a fundamental role for machines to understand the content of the image, being remote sensing image captioning a very important part of this field.

Although it should be noted that another of the problems of current remote sensing image captioning models is that they have not yet been able to fully use the semantic information in the images and this has led them to suffer from an over-fitting problem induced by the small size of the dataset.[115].

Of the many existing methods for remote sensing image captioning, most follow the basic guideline of natural image captioning, although there are even a few that have achieved combined success using the characteristics of remote sensing imagery, which is going to be explained further ahead.

As a novelty, Shi et al. [116] managed to propose an object detection model for remote sensing images, which does not depend on the use of an LSTM structure. This model is mainly divided into two modules which correspond to the visual analysis of the images and the subsequent generation of text, respectively, where the first part, that is, the visual analysis, uses a fully convolutional network (FCN).

If this new FCN structure is compared with the old, more traditional structures, the CNNs, the new structure allows retaining the spatial information of the input image without taking into account its original size, in addition to allowing detection methods of objects the retrieval of this information in the captions of remote sensing images.

On the other hand, Wang et al. [117] focused on the creation of a method that allows the retrieval of the information initially used in semantic embeddings to measure the distance between the visual representations of images and the textual representations of collective sentences, to generate appropriate captions that are located close to the input image in the imaginary space created by the representations of these. The study proposed a multisentence captioning task to describe the remote sensing images with five captions and consider the complex distribution of ground objects in remote sensing images. A collective image caption is generated that represents the five image captions, by vectorizing all the words and concatenating them into a collective sentence representation in origin order. After that, the image is represented by a pretrained convolutional neural network, and its representation and the collective caption representation are mapped to the same space for the captioning task.

Qu et al. [118] combined the characteristics of high-resolution remote sensing images and their corresponding text information, to consider all the information possible. This approach creates a collection of words that contained the common information among all input image captions from the dataset and is jointly incorporated into a captioning model data to generate a determinate sentence that covers common contents. Instead of employing a naive recurrent neural network, a memory network is introduced to generate sentences, jointly with a novel retrieval topic recurrent memory network. The authors proposed recurrent memory networks to utilize the topic words (imported into architecture naturally and flexibly as a part of extensible memory cells) to overcome the shortcoming of long-term information dilution in RNN, also applying a CNN model on the top of memory networks to capture the relationship between the topics and the images, as a control signal for captioning task.

It was not until the work of Lu et al. [119] that attention mechanisms began to be added to the encoder-decoder models to make better use of the visual information captured from remote sensing images and therefore generate more accurate descriptions. Zhang [120] managed to improve the basic mechanism of attention, creating the idea of attention by attributes, where low and high-level characteristics are combined, and thus more complex sentences can be generated by taking advantage of remote sensing attributes. By introducing the attributes, the attention mechanism perceives the whole image while knowing the correspondence between regions and words, so it can influence the intermediate vector by assigning different weights to different areas of the image and therefore compensate for the deficiency of the basic attention mechanism in complex remote sensing images.

One of the great problems of remote sensing image captioning, which was and still is, is the lack of data since it is an area that is rarely studied and if it occurs, the information collected is usually private or restricted domain. Bo Qu et al. were responsible for creating two datasets designated as UCM-Captions and the Sydney-Captions, which are based on the UC Merced Land-Use dataset [121] and the Sydney dataset, respectively [122]. Latter Lu et al. proposed the creation of a new database called RSICD [123], which greatly expands the size of the data, providing images with greater diversity and smaller differences between classes, providing researchers with a valuable data resource for remote sensing vision and language research. These databases became the most relevant for training and evaluating methods for the remote sensing image captioning task.

One of the most widely used mechanisms, already mentioned above and from which not even models with remote sensing images escape, is the attention mechanism. This is mainly because this mechanism quite directly simulates how the human being receives visual information and is therefore widely used in the understanding of images, thus allowing more information to be captured from the scene and improving the stability of the images. generated sentences, in this way a new attention mechanism called attention to the scene, is proposed.

Although for other computer vision tasks, such as object detection, the most important part is the detection and identification of objects in the image, for Image Captioning it is very important to understand the relationship between objects, tone and image, since although there may be a lot of information in an image, the task of captioning must be to pay attention to the general and most outstanding parts of it.

A clear example of Remote Sensing Image Captioning that perfectly explains the points discussed above is military use. In this field, a computer must use the remote sensing image captioning system to automatically convert images captured from a drone or aircraft on the battlefield into text or relevant information and send it to a command centre or first line, thus achieving a faster transfer of information and allowing a rapid response to any type of risk – see Figure 2.28 [124].

Leaving aside the more military plane, in terms of civilian and daily use, the improvement of the remote sensing image captioning task can help in the exploration of the arias in a later natural disaster and in this way guide the emergency teams through the wreckage of the event and even help locate potential injuries [125].

Before continuing, it is worth highlighting the main differences between remote sensing images and natural images, so that later the characteristics of remote sensing images can be explained in detail. Starting from the base, remote sensing images are taken from an aerial view, that is, from above, through a drone, plane or satellite, while natural images are usually taken from a human perspective.

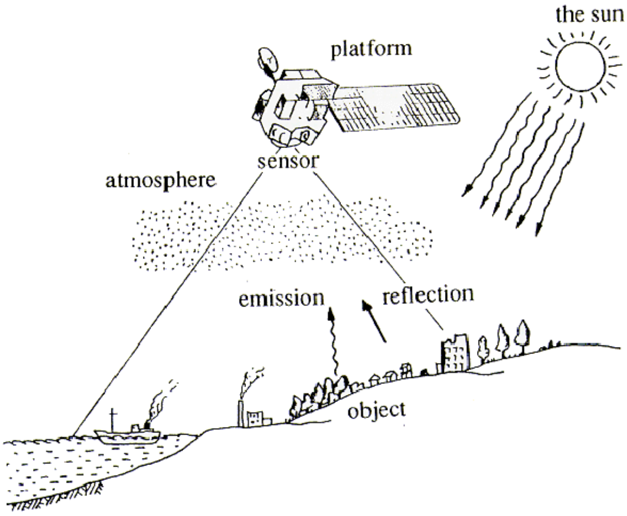


Figure 2.28: Remote Sensing Data Acquisition Example [23].

On the other hand, another noteworthy feature is that natural images are generally taken relatively close to the object, although a telephoto or long-range lens may be used, which compared to images obtained through remote sensing is relegated to a close plane, since the distances are too far in scale.

This indirectly implies that the detection area of the remote sensing image is much larger than the natural image and that the elements of the image will be found on a much smaller scale and may even become invisible [126].

One of the main things that must be made clear is that the task of captioning for remote sensing images becomes much more complex than the one that only requires captions for natural images [127], this fact is mainly because the images captured from remote sensing are taken from the ”point of view of God” which characterizes them for having much more ambiguity in their elements. A clear example of this problem is found when trying to make complex descriptions by humans in remote sensing images captured from planes, drones or satellites since the elements to be described lack certain characteristics that allow us to quickly recognize them [128].

These are the main challenges that characterize remote sensing images and that make them a much more complex task than it might seem at first: [129]

1) Ambiguity of scale: Terrestrial objects in remote sensing images can present different semantics at different scales, which causes when changing the scale from a top view, the elements that make up the image to be directly affected. Mainly this is also because there is no approach like that of natural images and therefore the system is seen in the need to describe all the important things in a remote sensing image, making it difficult to choose which information is more relevant than another. or which could have disappeared due to these special characteristics of remote sensing images.

2) Category ambiguity: Due to the large number of elements that an image taken from remote sensing can encompass, it is very difficult to effectuate an individual analysis of all the elements, since this would lead to an infinite number of categories for certain regions of images of remote sensing At the same time, there is also the problem that among this mixture of categories, the elements can merge at

a visual level and therefore make the identification task very complex and expensive since it would be difficult to identify them with a single category label.

A clear example of this point can be vegetation, plants usually tend to have a greenish colour and in a remote sensing image they could cover a large area of it, however, the main problem is that green vegetation includes green grass, green crops, green trees and bushes... and this makes it difficult to distinguish between each other when trying to identify the elements individually.

3) Rotation ambiguity: The very characteristic way of capturing remote sensing images, being taken from ”God’s point of view”, means that they do not have a specific direction, since there is no perspective and therefore the image can be modified infinitely at any angle of rotation or translation, since there is no perceived up, down, left or right in a remote sensing image.

The example for this case is quite simple and only requires a glance for the sale, in natural images, the elements always maintain a fixed direction and similar in most cases, where a building will always be built from below up and a person will be standing on the ground with his feet, we can hardly see a person with his head on the ground and his feet up.

As we have been able to observe throughout this section, remote sensing images have become an important tool for people to access geospatial information, we can see more and more examples of these images in daily life, where without realizing account, in recent years, these images are playing an increasingly important role not only in the military or private area but also in the field of commercial applications and finance.

The applications of these images already cover a wide range of vital areas, such as population censuses or counts in concentrations/demonstrations, exploration of remote areas and detection of materials (such as water, iron or even oil), mapping with real updated information on cities, environmental tests and changes, as well as the prevention of natural disasters or the timely evacuation of possible victims, the correct planning and subsequent location of railways and urban routes, archaeological research, the evaluation of the quality of the land, the detection of large volumes of traffic, traffic jams or accidents, among others – see Figure 2.28 [130].

Increasingly and more effectively, remote sensing images are taken with high-resolution devices, which makes their interpretation and understanding easier, although they remain truly limited at the level of features, such as scene classification and object detection with little reasoning and scene understanding until an efficient way is found to solve the ”semantic gap” problem between low-level features and high-level summary.

Turning the ”semantic gap” problem into one of the most challenging scientific problems in the field and once solved, it will allow the correct interpretation of high-resolution remote sensing images at different levels in a large dataset [131].

2.2.3 Captioning Geo-reference Images

In general, it is difficult to generate descriptions that contain references to the geographical context of an image, when speaking of geographical references, it refers to obtaining information near the location where a photograph is taken and/or relevant geographical objects. that are placed around the location of an image and therefore provide extra knowledge about the geographical context of the scene.

A recent study [24] proposes a way to build a geographic context image-specific representation and adapt the caption generation network to produce appropriate geographic names in image descriptions, where the data used contains contextualized captions and geographic metadata. This model, along with geographic metadata, enables the system to have the ability to generate geo-aware captions to produce image descriptions that include meaningful and contextually relevant geographic information.

In this way, we can say that to consider a geo-aware image captioning dataset, it must contain not only images with captions but also the geographic information related to the locations of the images or tags of relevant geographic elements that are used around the location – see Figure 2.29.

On the other hand, we must clarify that a geographic context is considered to be the set of relevant geographic objects around the location of the image, which are not only used to compile a specific vocabulary of images of geographic names, but also allow to complement the representation of geographic names images used by the text generation network [132].

As mentioned before, unlike semantic segmentation and instance segmentation, which are only capable of recognizing geographic objects separately, that is, individually, image captions can provide more complex semantic information, since they can learn and express the spatial relationship between these elements of an image [133].

This improvement in image captioning, in which geographic metadata is directly included, can directly help with searching, categorizing, and browsing through files, because the input image description can be used as a text feature in image indexing and thus allow photos associated with certain geographic coordinates to be found quickly and efficiently through an image search engine. Another point in favour lies in the fact that the current GPS photo recommendation is based mainly on the proximity of the location and therefore the generated caption could permit the rest of the image recommendation services to link the photos in terms of spatial connections and semantics [134].

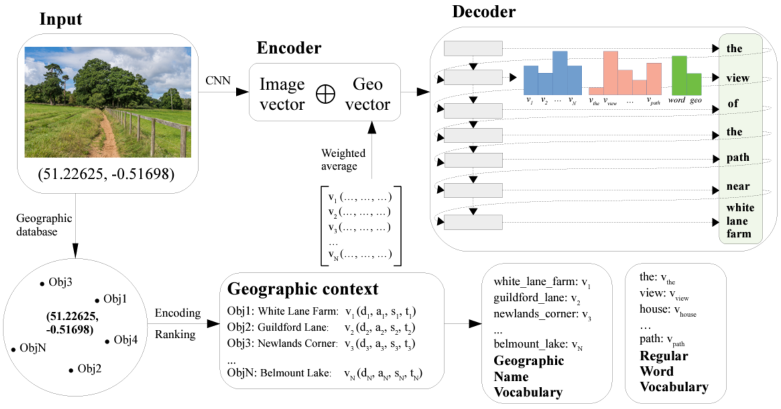


Figure 2.29: An overview of the geo-aware system architecture [24].

2.3 Overview

Throughout this chapter, the fundamental bases have been laid out to understand the work that has implemented and that is proposed with this project. In the first section, the basic knowledge about Artificial Intelligence and specifically the field of Deep Learning has been addressed, since the technology that has been used, as well as its internal operation from the simplest element, such as it, can be the perceptron, passing through the different neural architectures such as CNN or RNN, ending with much more complex networks such as the encoder-decoder architecture and the Transformer.

On the other hand, in the second section, the Image Captioning task has been proposed, its evolution over the years has been explained and what is its scope of application in different areas, due to the innovation that provokes by focusing both on the visual coding of images, and on the generation of language, that is, the captions. Once the general task was raised, the specific case was addressed, the Remote Sensing Image Captioning, for which all its special characteristics were explained that make it a much more complex task than those that only use a natural image, as well as gave a temporary context to its evolution and explained the different advances in architectures, ending with future applications.

Last but not least, the idea of using metadata was introduced, in this case, geographic information, as a measure to provide extra help to systems in the field of geolocation, as well as the possible opening of a large number of applications due to this possible external inclusion of information.

To finish and as a link measure for the next chapters, it is necessary to present the Table 2.1 with the results of other proposals made by studies about Remote Sensing Image Captioning, used as metrics for the evaluation BLEU-1, BLEU-4, METEOR and CIDEr and in this way, be able to have a global vision that allows us to compare the performance of previous models.

	Sydney				**UCM**				**RSICD**			
Method	BLEU-1	BLEU-4	METEOR	CIDER	BLEU-1	BLEU-4	METEOR	CIDER	BLEU-1	BLEU-4	METEOR	CIDER
CSMLF [117]	0.5998	0.3433	0.2475	0.7555	0.4361	0.1210	0.1320	0.2227	0.5759	0.2217	0.2128	0.5
RTRMN (statistical) [118]	—	—	—	—	0.8028	0.6393	0.4258	—	0.6102	0.2859	0.2751	—
Multi-Scale Cropping [135]	0.6150	0.4000	—	—	0.5940	0.4290	—	—	—	—	—	—
Soft-Attention [128]	0.7322	0.5820	0.3942	2.4993	0.7454	0.5250	0.3886	2.6124	0.6753	0.3617	0.3255	1.9643
Hard Attention [128]	0.7591	0.5258	0.3898	2.1819	0.8157	0.6182	0.4263	2.9947	0.6669	0.3407	0.3201	1.7925
FC Attention + LSTM [120]	0.8076	0.5544	0.4099	2.2033	0.8135	0.6352	0.4173	2.9958	0.7459	0.4574	0.3395	2.3664
SM Attention + LSTM [120]	0.8143	0.5806	0.4111	2.3021	0.8154	0.6458	0.4240	3.1864	0.7571	0.4612	0.3513	2.3563
Structured attention [136]	0.7795	0.5861	0.3954	2.3791	0.8538	0.7149	0.4632	3.3489	0.7016	0.3934	0.3291	1.7031
Cross-hierarchy attention [137]	0.817	0.591	—	2.291	0.823	0.659	—	3.192	0.770	0.471	—	2.363
SD-RSIC ResNet50 [130]	0.7160	0.3980	0.3200	—	0.7430	0.5150	0.3580	—	0.6490	0.2950	0.2490	—
SD-RSIC DenseNet169 [130]	0.7300	0.4670	0.3410	—	0.7470	0.5180	0.3750	—	0.6430	0.2850	0.2440	—
SAT (LAM-TL) [32]	0.7425	0.5369	0.3700	2.3563	0.8208	0.7229	0.4880	3.7088	0.6790	0.4148	0.3298	2.6672
Adaptive (LAM-TL) [32]	0.7365	0.5348	0.3693	2.3513	0.8570	0.7430	0.5100	3.758	0.6756	0.4077	0.3261	2.6285
ML Attention + Semantic [22]	0.8233	0.6003	0.4202	2.3110	0.8330	0.6623	0.4371	3.1684	0.7597	0.4623	0.3543	2.3614
Denoising-based fusion [138]	0.8324	0.5851	—	3.8198	0.8306	0.6345	—	3.2956	—	—	—	—
VRTMM+SCST [30]	—	—	—	—	—	—	—	—	0.7934	0.5113	0.3726	2.7930
Multi-level attention [1]	0.7900	0.6052	0.4741	2.1811	0.8864	0.7271	0.5222	3.3074	0.8058	0.5163	0.4718	2.7716
Continuous Representations [139]	—	—	—	—	0.8510	0.6666	0.4229	3.4239	0.7846	0.5190	0.3714	2.7777

Table 2.1: Overview of previous results for remote sensing image captioning

Chapter 3

Proposed Approach

This chapter will expose and explain the concepts that describe the architecture or rather the proposed method, for the improvement of the captioning task of remote sensing images, covering both the technical details of the creation of the method as well as the reasons why the different variations in the architecture and structure of the presented model have been chosen.

The first Section 3.1 describes the resources and techniques that have been used to implement the Contrastive Language-Image Pre-training (Contrastive Language-Image Pre-training (CLIP)) model for the remote sensing images task and provides additional details about its implementation, the advantages that this has and the possible problems in its use, due to the adaptation of the previously pretrained model. While in the second Section 3.2, the aspects related to the real implementation of the CLIP model for the image captioning task are addressed, thus completing the proposal for this model.

Finally, in the Section 3.3, the implementation details of the proposal will be fully described with a detailed description of the main components that make up the architecture, thus allowing a correct understanding of the model proposed in this work, accompanied by images that illustrate its structure and therefore can facilitate the full understanding of the presented scheme.

3.1 Using CLIP for Remote Sensing Imagery

The CLIP model, Contrastive Language-Image Pre-training [25], is initially based on a large number of studies implemented in the fields of zero-shot transfer, natural language supervision and multimodal learning, finding the origin of these projects and their research decades away [140], but with the problem of little progress in technology. It was therefore not until more recent years that, computer vision technology was studied and used solely as a way of generalizing to categories of invisible objects [141, 142], thus being a completely innovative idea the fact of taking advantage of natural language as a flexible prediction space to allow generalization – see Figure 3.1.

Therefore, and based on the research of Richer Socher, together with the authors of the Stanford platform [143] in 2013, a model was developed in Canadian Institute For Advanced Research (CIFAR)-10 [144] that allowed or better said, could make predictions in an embedding space of word vectors, thus showing that this model could predict two invisible classes.

Appearing that same year, an extension of the approach, where Deep Visual-Semantic Embedding Model (DeVISE) [145], found that it was possible to fit the ImageNet model in such a way that it could be generalized to correctly predict objects outside the original training set.

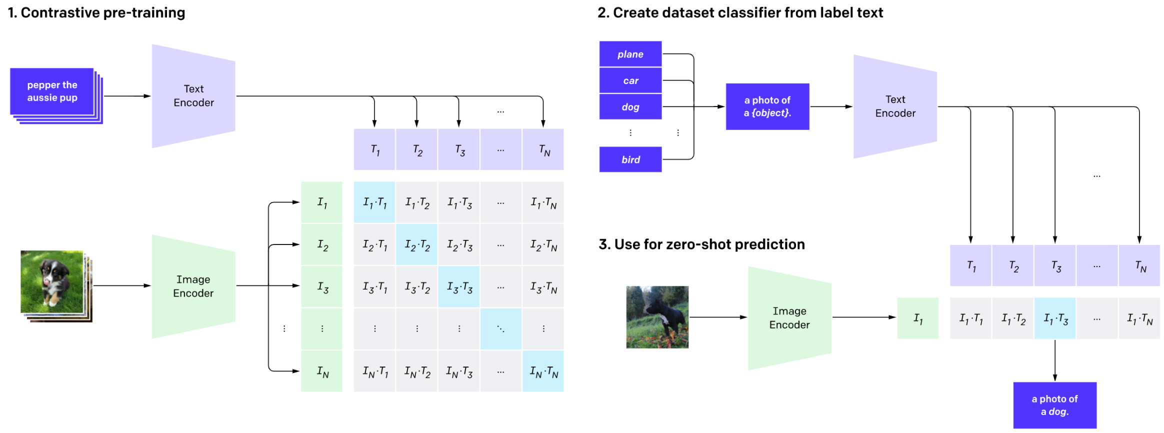

1. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

Figure 3.1: Summary of the original approach of CLIP architecture [25].

One of the most inspiring works for the creation of this architecture is that of Ang Li at Facebook AI Research (FAIR) [146], which demonstrated in 2016 that the use of natural language supervision allowed the transfer without triggering several datasets of existing computer vision classification systems, such as the already established dataset known as ImageNet. This proposal was mainly based on a CNN that allowed predicting a much broader set of visual concepts, counting on visual n-grams, from the text of titles, descriptions and labels obtained from the Flickr dataset, which has around 30 million photos, achieving a result of 11.5% accuracy on ImageNet.

In this way, CLIP is part of the set of proposals for learning visual representations under the supervision of natural language, where more modern architectures are usually used, such as the Transformer [87], where we find other architectures such as Visual Representations from Textual Annotations (VirTex) [147], which is based on autoregressive language modelling, Image Conditioned Masked Language Modeling (ICMLM) [148] in masked language modelling, and Contrastive VIsual Representation Learning from Text (ConVIRT) [149] that studies the same scope of the CLIP proposal but in the medical field.

In this way, CLIP presents a neural network architecture, which can learn visual concepts efficiently from natural language supervision, this model can apply to any visual classification reference point simply by providing the names of the visual categories that will be recognized, this operation being very similar to the ”zero trigger” method of architectures such as Generative Pre-trained Transformer (GPT)-2 and GPT-3 [150, 151].

In addition to the basic tasks, this architecture model addresses some of the big problems that the field of deep learning for computer vision has been facing. Therefore, problems such as the more typical vision datasets are labour-intensive and therefore costly to create, while only teaching a sometimes very limited segment of visual concepts as well as the inconvenience that standard vision architectures or models are usually very effective in a given task, but only in a single task, and the fact of trying to adopt or implement the models in a new task requires a significant and enormous effort.

They are addressed and resolved since the network is trained on a wide variety of images in conjunction with a wide variety of natural language monitoring that is abundantly available and accessible to all on the Internet, which not only allows it to be optimized for many visual concepts but also to be effective in many possible areas or fields of application.

In turn, the fact that models that perform well in benchmarks in the first place often perform disappointingly poorly in stress tests [152, 153, 154, 155] is addressed, since by design the model is capable of receiving natural language instructions to perform a large variety of classification benchmarks, without directly optimizing the performance of the benchmark, this being a very important point since by not optimizing directly for the benchmark, the result becomes much more representative, achieving results similar to the original ResNet-50 [74] and solving the ”robustness gap problem” by up to 75%.

Therefore, the CLIP model achieves precision values similar to the most advanced models, but obtains much more representative performance results than the rest, when performing the evaluation using different configurations that differ from ImageNet, such as ObjectNet [156], which checks a model’s ability to recognize objects in many different poses and with many different backgrounds inside houses or as ImageNet Rendition and ImageNet Sketch [157, 158], which checks a model’s ability to recognize more accurate representations of abstract objects.

In this way, although initially at the time of the creation of the CLIP architecture, different models for the image encoder were evaluated, one of these architectures being the ResNet-50 [74], mainly due to Widespread adoption and proven performance and using improved versions such as ResNetD [159] and Zhang’s Res-2 blur pooling antialiasing [160]. The architecture finally chosen was based on a Vision Transformer (ViT), a model recently presented by Dosovitskiy in the year 2020 [89], modifying only the addition of an additional normalization layer to the combined patch and the position embeddings before the Transformer and using hence a subtly different initialization scheme – see Figure 3.2.

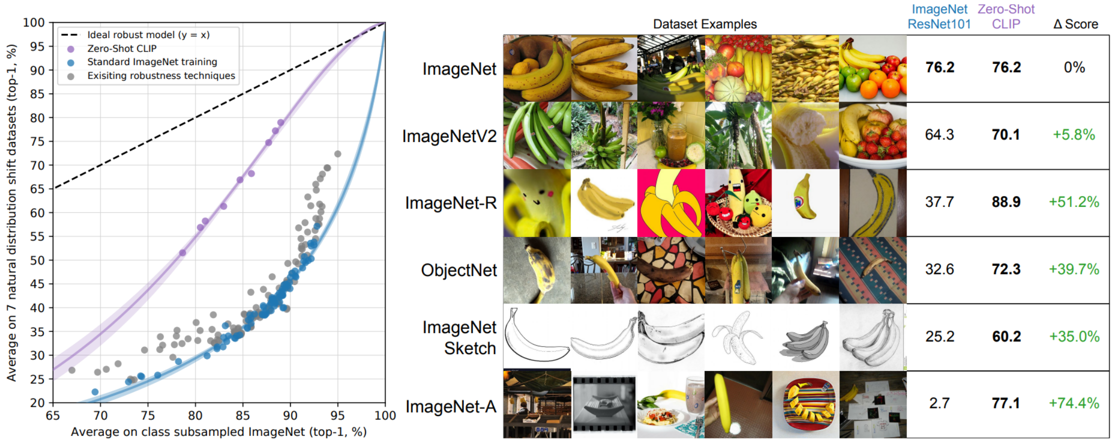


Figure 3.2: Performance of zero-shot CLIP model (ViT-L/14) compared with the ResNet-101 [25].

Therefore, the text encoder is a Transformer [87] with the architecture modifications described in [150] with a total of 63M parameters, 12 layers that included 8 attention heads, operating on a lowercase byte pair encoding (BPE) [161].

The text input sequence is enclosed with Start Of Sequence token ([SOS]) and End Of Sequence token ([EOS]) tokens. The activations of the highest layer of the Transformer in the [EOS] token are treated as the representation of features text that is normalized into layers and then linearly projected into the modal embedding multi-space. The masked self-attention is applied directly in the text decoder to preserve the ability to initialize with a pre-trained language model, as well as to add language modelling as an auxiliary target.

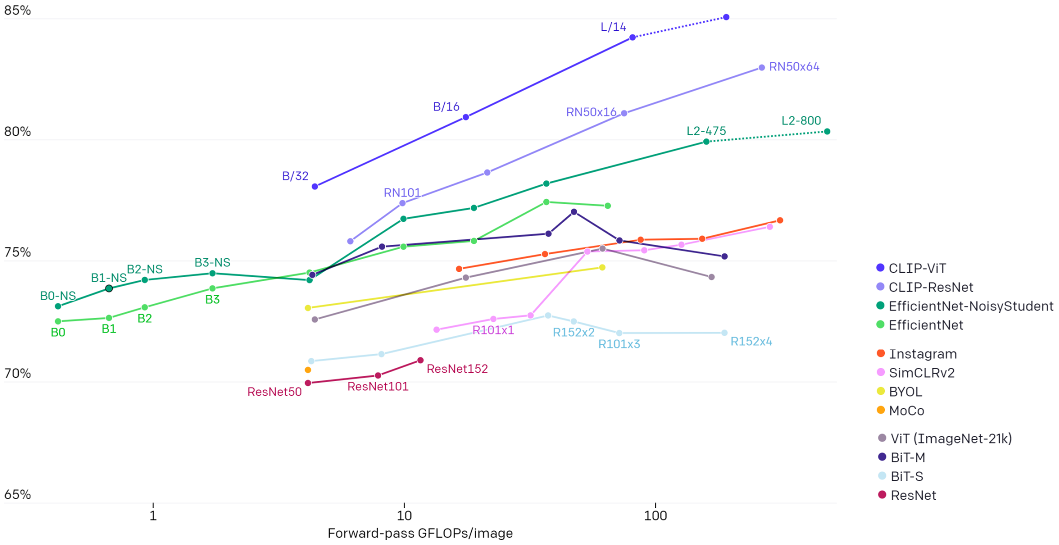


Figure 3.3: Average linear probe score across 27 datasets [25].

Thus summarizing the CLIP model as a multimodal language and vision architecture, which can be used both for image and text similarity and for zero-trigger image classification, where a similar Transformer to ViT is used to obtain visual characteristics and a causal language model to obtain the text characteristics and in the same way, both characteristics are projected in a latent space with identical dimensions, where a scalar product between the projected image and the text characteristics is calculated which is later used as a similar score for homework – see Figure 3.3.

Its operation is reduced to the sending or transmission of images to the Transformer encoder, where each of these images is divided into a sequence of patches of fixed size, which do not overlap and which are subsequently linearly embedded, to which is added a token Classification token ([CLS]) to serve as a representation of a complete image. In turn, absolute position embeddings are also added and the resulting vector stream is fed into a standard Transformer encoder.

For the specific case of using the CLIP model set for remote sensing images, the images of the RSICD test division have been previously encoded and these encodings have been stored in a recovery based on Non-Metric Space Library (NMSLib) approximate nearest neighbour.

This allows a text-to-image search, where a text feature that describes some natural or artificial geographic feature, such as ”beach,” ”mountain,” ”school,” or ”baseball field,” is encoded and compared with the NMSLib image encoding index, returning images with vectors with more cosine similarity compared to the query vector. Also, an image-to-image search is allowed, where an image is encoded and compared to the NMSLib image encoding index, returning images with more similar vectors.

3.2 Adapting CLIP for Image Captioning

To accurately explain the adaptation of the CLIP model in the Image Captioning task, the new architecture for this image captioning task must first be detailed, from the perspective of sequence-to-sequence prediction, which is called Caption TransformeR (CPTR) [26]. This new type of model, compared to the traditional CNN + RNN design architecture, takes raw images as input for Transformer, thus being able to obtain the global context in each layer of the encoder from the beginning, without the need to make use of no convolution, in addition to providing own attention between patches in the encoder and the attention of ”words-to-patches” in the decoder.

This new approach to work is completely inspired by previous work in the field and provides a new perspective, where a complete network of Transformers is used to replace the traditional CNN structure in the encoder part with a fully convolutional Transformer encoder. If we compare this new model with the more conventional image captioning architectures, we see how these take as input the function extracted by CNN or the object detector, while the CPTR directly sequence the unprocessed images as input, dividing the image in small fixed-size patches (e.g., 16 × 16), flattening each of these patches and reshaping them to build a sequence of 1D patches. Once the sequence is formed, it is sent to a patch embedding layer and a positional embedding layer so that relevant information can be extracted before being introduced into the Transformer encoder.

As you can see, this type of architecture is much less complex than the CNN + RNN paradigm, but in turn produces much more effective results, as in this way you can completely avoid any convolution operation, an operation that as already known as the CNN encoder, it is subject to a limitation in global context modelling that can only be solved by gradually expanding the receptive field as the convolution layers are deeper. While not only does CPTR lack these limitations, it is also able to create long-range dependencies from the beginning between sequential patches by leveraging the use of the self-care mechanism, because, during word generation, attention is modelled in the cross-focus layer of the decoder – see Figure 3.4.

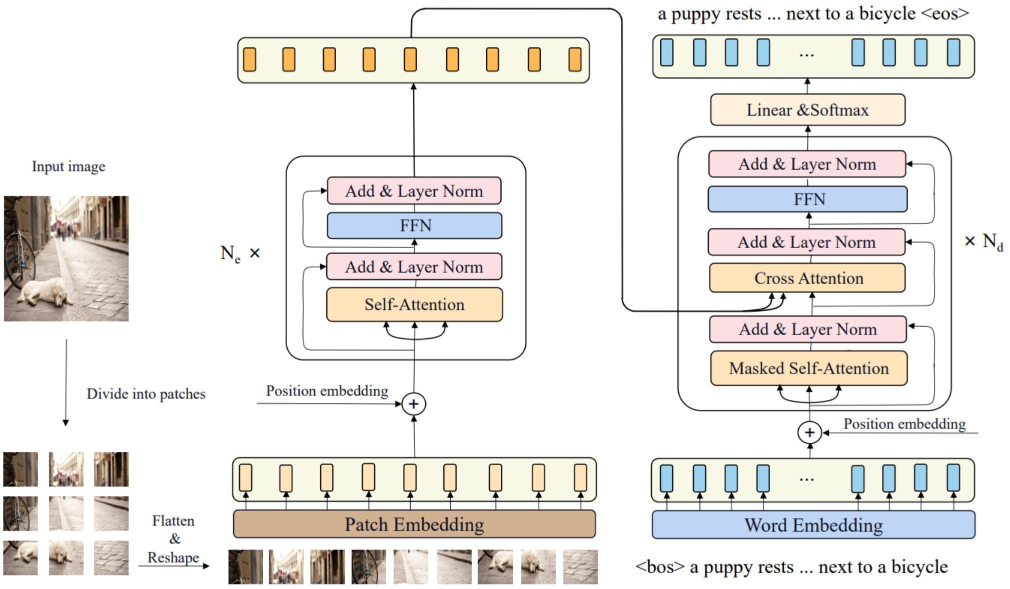


Figure 3.4: The overall architecture of proposed CPTR model [26].

Going into a little more detail in the structure of the CPTR, we can extract two large blocks, first, the encoder the module that is responsible for extracting spatial features of the images, which is different from previous architectures where it was used a pre-trained CNN model o Faster Region Based Convolutional Neural Networks (R-CNN), in this case, the input image is sequenced and the captions of the image are treated as a sequence-by-sequence prediction, whereas already mentioned, the original image is divided into a sequence of patches of fixed size for each image and thus allows a specific adaptation to the input form of Transformer.

On the other hand, the decoder is a module in which a sinusoidal positional embedding is added to the word embedding features and the results of the sum and output characteristics of the encoder are taken as an input. In this particular architecture, the decoding is based on a certain number of identical stacked layers, where each of these layers contains a masked multi-head self-care sub-layer, which is followed by a multi-head cross-care sub-layer and a sub-layer of positional feedback sequentially, thus allowing to provide global context information in each layer of the encoder from the beginning.

As already mentioned, the image captioning task is a fundamental task in visual language understanding, where the given model or architecture is capable of predicting an informative textual caption for a given input image. Using specifically, the recently proposed CLIP model, which contains rich semantic features, since it has been previously trained with textual context, thus making it one of the best models for vision and language perception, thus obtaining a comprehension of visual and textual data.

In this specific case, the CLIP model used is designed to impose a shared representation for both images and text messages, since when trained with a large number of images and textual descriptions using a contrastive loss mechanism, their visual representations and textual are incredibly well correlated, this correlation is a fact that saves both training time and data requirements.

One of the most innovative approaches in which the CLIP model is used together with a GPT-2 architecture, corresponds to the ClipCap proposal [162]. This recently proposed CLIP model contains rich semantic features since the model has been trained with textual context, making it best for vision-language perception. The main idea of the proposal is the use of CLIP encoding as a prefix to the caption, by employing a simple mapping network, and then together with a pre-trained fine-tune a language model (GPT2) to generate the image captions, being able to obtain a wide understanding of both visual and textual data.

In detail, the method exposed produces a prefix, a fixed size embeddings sequence, for each caption when applying a network of mapping onto the CLIP embed, where the produced prefix refers to a sequence of fixed-size embeds, which is concatenated with the caption embeds, these being in turn fed by the language model, which is adjusted together with the mapping network training. This scheme is mainly based on a transformer architecture because it allows for successfully producing meaningful captions while imposing substantially less trainable parameters. This very innovative proposal, since it reduces the aforementioned gap between the visual and textual worlds, and therefore allows the use of a simple mapping network.

This approach is less computationally demanding, where only the mapping network is entered, while CLIP and the language model are kept frozen, are inspired by Li et al. [163], a proposal in which it is shown that it is possible to modify and adapt a mainly language model, for new tasks, through the use of the concatenation of a prefix learned or produced by a similar model, thus achieving to demonstrate the generation of rich, diverse and meaningful texts, as well as a good generalization for complex scenes.

This ClipCap approach has been able to, using a shorter training time, achieve comparable results to more advanced approaches on challenging conceptual [164] and NOCAPS [165] caption datasets, and marginally lower for the COCO more restricted [166, 167] benchmark, as well as being able to provide a comprehensive analysis of the length of the required prefix and the effect of adjusting the language model, including also the interpretation of the prefixes produced.

The main challenge of the ClipCap study is to translate between the representations of CLIP and the language model, because their semantic spaces were independent, as they were not jointly trained, even though both models develop a rich and diverse representation of text. The authors propose fine-tuning the language model during the training of the mapping network, thus allowing a more expressive outcome, as well as additional flexibility for the networks. Although this is a great solution, they also detail in their study an additional variant of the approach, in which they keep the language model fixed during the all training process.

The proposed approach essentially learns to adapt existing semantics of the pre-trained models to the target dataset, instead of learning new semantic entities, being relevant to emphasize the utilization of these powerful pre-trained models and the understanding of how to harness their components.

3.3 Implementation Details

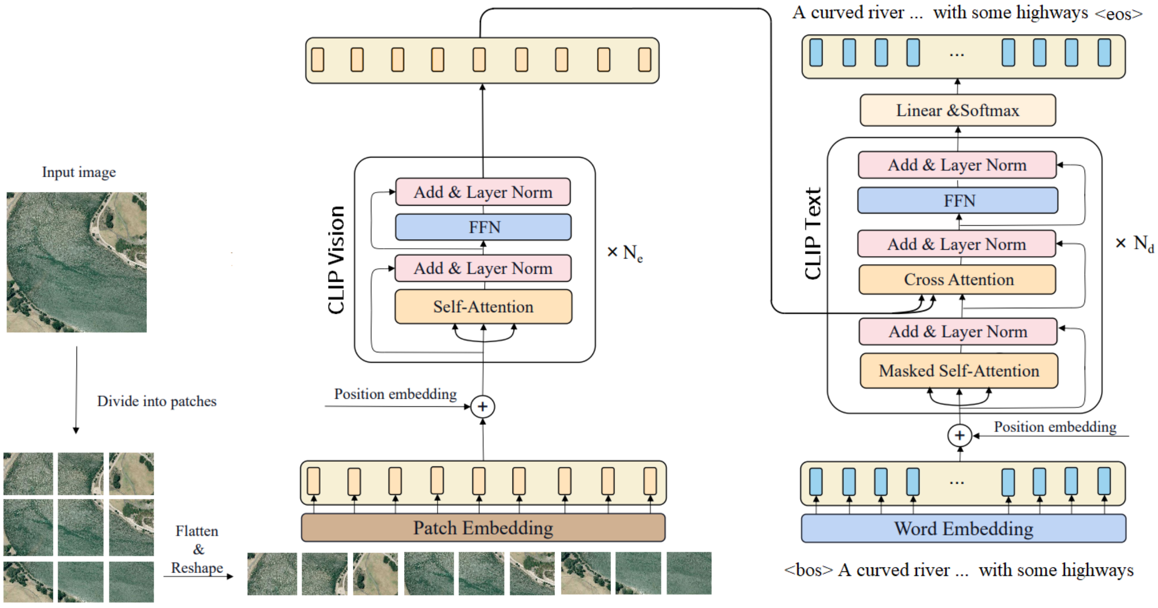


Figure 3.5: Illustration for the proposed full Transformer-based encoder-decoder architecture using a fine-tuned version of CLIP for image captioning.

The structure of the proposed Transformer-based encoder-decoder model method is shown in Figure 3.5 where both the image encoder and the decoder, which consist of a Transformer model, use fine-tuned components from a CLIP model adjusted to the RSICD dataset for caption generation.

The proposed model structure is adapted from the CPTR proposal, where a full Transformer network is used to overtake the traditional CNN + RNN design architecture in which the CNN in the encoder part is replaced with a Transformer encoder. In our proposal, the CLIP image encoder is a ViT encoder adapted from the CLIP model, specifically from CLIP-RSICD an adjusted version to the RSICD dataset for caption generation. On the other hand, our CLIP text component is also an adapted Transformer, similar to BERT, that allows us to initialize the decoder weights.

By using a full Transformer network structure using pre-trained models, not only do we lack the limitations of traditional architectures for image captioning, but the model can create long-range dependencies from the beginning between sequential patches by leveraging the use of the self-care mechanism, because, during word generation, attention is modelled of ”words-to-patches” in the cross-focus layer of the decoder.

The model used, called CLIP-RSICD, is available for use on HuggingFace Models. This is a fine-tuned model that can be used in the same way as the original CLIP model, where it learns to project images and text onto a common embedding space such that similar (image, image), (text, image), and (text, text) pairs appear closer in this space than dissimilar pairs.

The CLIP model is a multimodal language and vision architecture, which can be used both for image and text similarity and for zero-trigger image classification, where a ViT is used to obtain visual characteristics and a language model is used to obtain the text characteristics in the same way. Both characteristics are projected in a latent space with identical dimensions, where a scalar product between the projected image and the text characteristics can be calculated, and later used as a similar score.

For images and text to be connected, they must both be embedded. In this case, the text encoder and image encoder get fit by simultaneously maximizing the cosine similarity when the text and image coincide and minimizing the cosine similarity when they don’t align, across all of our text+image pairs. Once all the model is fit, a new input image into the image encoder should retrieve the text caption that best fits the image – or, vice versa.

The CLIP-RSICD model was based on a CLIP model released by OpenAI and further adjusted on the existing remote sensing image captioning datasets using the Adafactor [168] and Adam [169] optimizers, with a learning rate of 5e-5 and a linear learning rate schedule.

All the information regarding the script used for fine-tuning the CLIP model on a TPU Virtual Machine and a similar script that can be used to reproduce the training can be found in the Google Cloud Platform (GCP) and the GitHub repository[1] accompanied by Colab Notebooks and the evaluation reports.

Mainly because the available datasets (RSICD, UCM-Captions, Sydney-Captions) for remote sensing were fairly small, the authors used an augmentation strategy. Both image augmentation and text augmentation were used to regularize the training and prevent overfitting.

[1] https://github.com/arampacha/CLIP-rsicd

While image augmentation was done inline using built-in transforms from Pytorch's Torchvision package, like random cropping, random resizing and cropping, colour jitter, and random horizontal and vertical flipping, text augmentations to image captions were done offline via back-translation using the Marian MT [170] family of translation models, specifically the ROMANCE models from Helsinki-NLP [171]. Each augmentation corresponded to back translation through a different pair of language models.

In a traditional encoder-decoder architecture for image captions, the image encoder is responsible for mapping the image inputs into real value vector representations. Specifically in remote sensing image captioning, the image encoder is typically a CNN model previously trained on the ImageNet dataset [172], which would be used to extract spatial features or features at ground level, which differ a lot from those of remote sensing image [1]. In this way, we choose to sequential the input image and treat image captioning as a sequence-to-sequence prediction task.

The architecture finally chosen for the CLIP image encoder was an adapted version of the CLIP model, based on a Vision Transformer (ViT), a model recently presented by Dosovitskiy in the year 2020 [89], with only the addition of an auxiliary normalization layer to the combined patch and the position embeddings before the transformer, and using hence a subtly different initialization scheme.

A division is made of the original input image, to convert it into a sequence of image patches, this way adapting it to the Transformer input form. The size of the input image is modified into the fixed resolution, taking into account the 3 colour channels. Then the already resized image is divided into *N* patches of fixed size, where *P* is the patch size, using a kernel and stride size of 32. Each one of these will be flattened and reconstructed to form a 1D patch sequence. Finally using a linear keying layer, the flattened sequence is mapped to latent space and a 1D positional key is added to the patch features.

$$\mathrm{X} \in \mathrm{R}^{H \times W \times 3} \qquad N = \frac{H}{P} \times \frac{W}{P}$$

The CLIP image encoder consists of 12 identical stacked layers with a feature dimension of 768, each of which is made up of a multi-head self-attention (MHA) sublayer followed by a positional feedforward sublayer. In the case of *MHA*, it contains *H* parallel heads, specifically 12 heads, where each of these *hi* corresponds to an independent scalar product attention function that allows the model to be able to jointly pay attention to different subspaces, following which a linear transformation W^{O} is used so that the attention results of the different heads are added.

$$\mathrm{MHA}(Q, K, V) = \mathrm{Concat}(h_1, \ldots, h_H)W^O.$$

The scaled dot product attention is a particular attention proposed in the Transformer model, where $\mathrm{Q} \in \mathrm{R}^{N_q \times d_k}, \mathrm{K} \in \mathrm{R}^{N_k \times d_k}$ and $\mathrm{V} \in \mathrm{R}^{N_k \times d_v}$ are the query, key and value matrices, respectively and where a scalar product is first calculated for each query and then each result is divided and a softmax function is applied.

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V.$$

The followed positional feed-forward sublayer is implemented as two linear layers with the GELU activation function, and with the dropout method between them to further transform features.

$$\mathrm{FFN}(x) = \mathrm{FC}_2(\mathrm{Dropout}(\mathrm{GELU}(\mathrm{FC}_1(x)))).$$

Finally, in each of the sub-layers, a sub-layer connection is placed, which is composed of a residual connection, together with a layer normalization, where the sub-layer can be an attention layer or an advance layer indifferently.

$$\mathrm{X}_{out} = \mathrm{LayerNorm}(x_{in} + \mathrm{Sublayer}(x_{in}))).$$

The CLIP text component, used to initialize the decoder weight in our model, is a Transformer [87] similar to BERT. We use this model together with the initialization of cross-attention randomly, meaning the attention mechanism in Transformer architecture that mixes two different embedding sequences is arbitrarily defined. The architecture is based on the modifications described in [150] with a total of 63M parameters, 12 stacked layers, a feature dimension of 512, including 12 attention heads operating on a lowercase byte pair encoding (BPE) representation of the text with a vocabulary size of 49.152 and limited by computational efficiency of the maximum sequence length of 76 [161].

The decoder adds a sinusoidal positional embedding to the word embedding features and in this way, takes the sum results and the output features of the encoder as input. Similar to the encoder, the decoder consists of several stacked identical layers, where each layer contains a masked multi-head self-attention sublayer followed by a multi-head cross-attention sublayer and a sequentially positional advance sublayer. The output function of the last layer of the decoder is the one in charge of predicting the next word through a linear layer whose output dimension is equal to the size of the vocabulary. Given a ground truth sentence $y^*_{1:T}$ and the prediction y^*_t of the captioning model with parameters θ, we minimize the following cross entropy loss:

$$\mathrm{L}_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y^*_t | y^*_{1:t1})).$$

Chapter 4

Experimental Evaluation

In this chapter, the concepts related to the experimental evaluation of the architecture proposal are implemented, covering both the technical aspects of the evaluation methods as well as the reasons why these have been chosen for conduct the correct validation of the presented model.

Being in the first Section 4.1, where the resources and techniques that will support the validation of the model are described, more specifically explaining the remote sensing image datasets datasets that were available for the training and evaluation of the system, as well as the metrics are chosen for its verification, including in the latter some interesting new metrics that are specified in these models. On the other hand, additional details about the experimental protocol will also be provided, this includes the description of the data pre-processing and post-processing, as well as the training hyperparameters that have been considered relevant for the final result.

Finally, in Section 4.2, the results obtained by using the proposed model are presented and compared with previous studies in the same field, as well as once the metrics have been presented in the previous section, a complete analysis of the possible capacities for improvement and new possible implementations, this explanation being complementary to the next chapter.

4.1 Datasets and Evaluation Metrics

In this way, to implement a correct evaluation of the proposed method and compare it with other methods, two main elements are required. First, one or several popular remote sensing image datasets to perform the experiments, both in terms of training and validation and final evaluation of the model, keeping in mind that these data must be correctly designed and structured for ease of use.

And as a second element, a correct choice of the metrics to be able to evaluate the proposed method and compare the results of different methods of image captioning is necessary for this second section to check which metrics have been used in the previous articles published in the field of application, as well as an investigation on some new metrics that can provide interesting data on the performance of the proposed model and that, in turn, adjust to the implemented standards.

4.1.1 Datasets details

As we have been seeing throughout the work, a large number of proposals have been made on different methods for captioning images, but most of these are specific to natural images and very few focus on the task of remote sensing image captioning, this is due in part to the great problem that there are many datasets for common objects, being one of the most relevant in the field ”Common Objects in Context” (Common Objects in Context (COCO)) [166], a large-scale object detection, segmentation, and captioning dataset with about 330K images (>200K labelled), which include features such as recognition in context, super-pixel stuff segmentation, 1.5 million object instances, 80 object categories, 91 stuff categories, 5 captions per image and 250,000 people with key-points, while for remote sensing images there are only a few small authoritative datasets.

Perhaps one of the biggest obstacles to the creation and application of remote sensing images is their specific characteristics, which differentiate them from natural images and which are necessary to take into account when creating/using a dataset. own self. One of the main factors to take into account is that these images can cover large regions, which in turn can contain multiple categories of land cover, a clear example being green areas where these can be from green plants, which will include grass green, green crops and green trees, which are difficult to distinguish from each other to swampy areas, mouldy swimming pools or even green roofs.

Another important factor is the amount of spatial information provided by the image, this must be taken into account, since remote sensing images will lack spatial characteristics such as perspective, scale and rotation and translation because for these images there is no difference between up and down, left and right and therefore there is no direction information.

Before explaining the remote sensing image datasets that have been used in the work, it is worth highlighting a specific dataset, perhaps one of the initial ones in the field of remote sensing images, which was proposed in this article [124], and that it does not have an official title since it was not disclosed, mainly due to the sentences in are more like a fixed semantic modal added in multiple object detection with examples like ”this image shows an urban area”, ”this image consists of some land structures”, ”there is a large aircraft in this image” which caused a lack of flexibility and diversity.

Taking into account the characteristics that have just been explained, the most recognized remote sensing image datasets are presented, their characteristics are explained and why they have been chosen for the evaluation of the system.

First, the UCM-captions dataset, proposed in ”Deep semantic understanding of high-resolution remote sensing image” [127], expanding the original University California (UC) Merced land use dataset [173], proposed in ”Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification” [121] where five detailed sentences have been added to describe the content of each image, with a huge inter-class diversity among the image groups, but at the same time with a small intra-class between images of the same group, creating a total of 10,500 descriptions – see Figure 4.1.

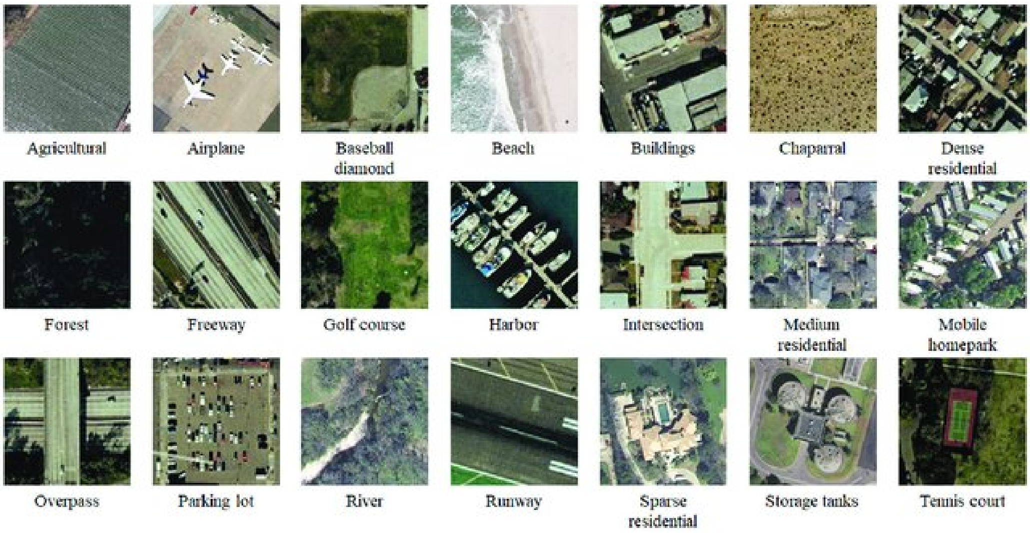


Figure 4.1: Example images of the UCM dataset [27].

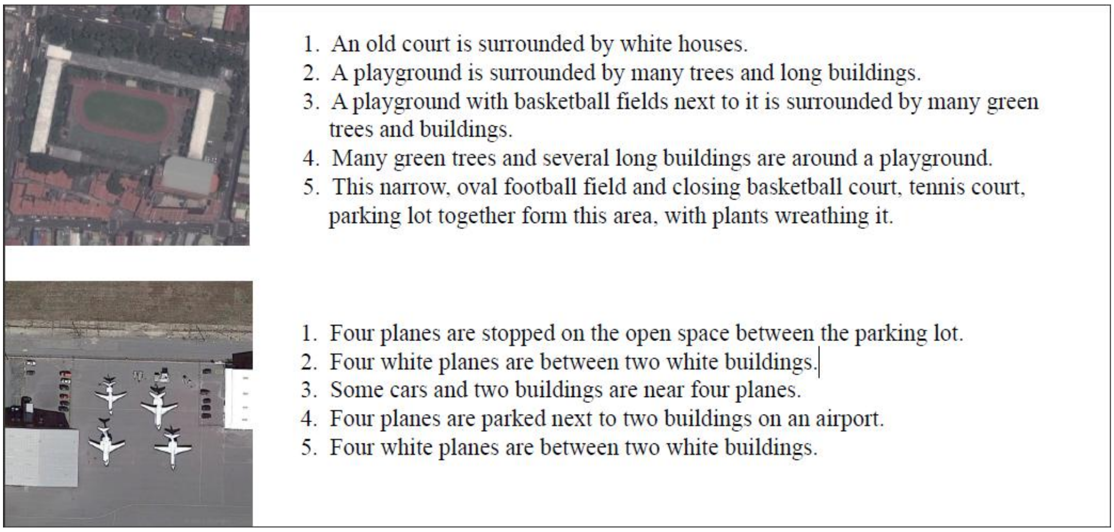
This set contains a total of 2100 RGB images, with a size of 256 × 256 pixels and a pixel resolution of 0.3048 m, where there are 21 land-use image classes, including agriculture, airplane, baseball diamond, Beach, Buildings, Chaparral, Dense Residential, Forest, Highway, Golf Course, Harbor, Intersection, Medium Residential, Mobile Home Park, Overpass, Parking Lot, River, Airstrip, Sparse Residential, Storage Tanks & Golf Course tennis, with 100 images for each class. The images in the UC Merced land-use dataset were manually extracted from many large images of the United States Geological Survey (USGS) National Map Urban Area Images [174].

Below we find the Sydney-captions dataset, which, as its name indicates, proposed in the article ”Deep semantic understanding of high-resolution remote sensing image” [127] and which is a version of a dataset of Sydney, as its name suggests, proposed in ”Saliency-Guided Unsupervised Feature Learning for Scene Classification” [122]. This dataset was obtained using Google’s platform, called Google Earth and is considered the smallest of the sets used since it only has a quantity of 613 RGB images and 5 captions per image. Each image has a size of 500 x 500 pixels, a ground resolution of 0.5 m per pixel and they are classified into 7 classes after cropping and selecting, these being: residential, airport, grassland, rivers, ocean, industrial and runway...

For both the UCM and Sydney-caption databases, one must appreciate that to be able to comprehensively describe a remote sensing image, one must pay attention to the different attention of different people to an image and different patterns of sentences, being important and also should be noted that these datasets only focus on the second point and the first point must be considered to be equal or even more important.

Finally, and perhaps most importantly, we find the dataset called RSICD originally proposed by Lu et al. [119] and taking as reference for its creation the work of natural image captions [175, 176, 167], which has a total of 10,921 RGB images that were compiled using sources such as Google Earth, Baidu Map, MapABC and Tianditu, with a size of 224 x 224 pixels, with 5 captions for each, generated by volunteers with experience in annotation and knowledge related to the field of remote sensing. In this generation, each volunteer is required to provide one or two sentences for a remote sensing image as a sample of diversity, following guidelines such as the obligation to describe all important parts of the remote sensing image, the prohibition to initiate the phrases with “There is” when there is more than one object in an image, the lack of use of the vague concept of words such as big, tall, many, in the absence of contrast, the prohibition of names of directions, such as north, south, east and west and the requirement of a minimum of six words per sentence.

In this way, a total number of 24,333 sentences were generated and a total of 3,323 words make up these sentences, being more detailed 724 images are described by five different sentences, 1,495 images are described by four different sentences, 2,182 images are described by three different sentences, 1667 images are described by two different sentences and 4853 are described by one sentence. Next, and to enrich the amount of information generated, the sentences were expanded to 54,605 sentences, randomly duplicating the existing sentences when there are not five different sentences to describe the same image – see Figure 4.2.



1. An old court is surrounded by white houses.
2. A playground is surrounded by many trees and long buildings.
3. A playground with basketball fields next to it is surrounded by many green trees and buildings.
4. Many green trees and several long buildings are around a playground.
5. This narrow, oval football field and closing basketball court, tennis court, parking lot together form this area, with plants wreathing it.

1. Four planes are stopped on the open space between the parking lot.
2. Four white planes are between two white buildings.
3. Some cars and two buildings are near four planes.
4. Four planes are parked next to two buildings on an airport.
5. Four white planes are between two white buildings.

Figure 4.2: Example images of the RSICD dataset [27].

Before continuing with the next section and after having presented the databases with which we have worked, it is important to review some important factors that affect both positively and negatively the results obtained. First of all, it should be noted that the datasets that have been used and, as has already been explained in each of the sets, do not correspond to their original versions, but rather are altered and modified versions, mainly because the initial sets had serious sentence problems, whether they were misspellings of words, grammatical errors, punctuation errors, or even errors in the conjugation of singulars and plurals.

This forced the original databases to be discarded and new ones had to be created from the modification and correction of the original ones since although errors as simple as singular and plural ones would be easily identifiable by a human evaluator, for a model created from artificial intelligence, which does not contain any grammatical, lexical or semantic correction module, it would be impossible to identify and correct them since the words are represented in the form of a vector or tokens. It is also important to comment that many of the words or vocabulary errors would be observed as common words since the error would only be removable when observing the complete meaning of the sentence or the understanding of the scenario and that therefore any attempt to pre/post-processing could trigger a severe loss of potentially important data, correcting these errors in the following article – see Table 4.1 [177].

	Modified Number/Total Number (%)		
Dataset	**Words**	**Sentences**	**Images**
Sydney	38/237 (16.03)	160/1865 (8.62)	123/613 (20.07)
UCM	44/368 (11.96)	374/10500 (3.56)	319/2100 (15.19)
RSICD	498/3325(14.98)	7166/54,605(13.12)	2493/10,921(22.83)

Table 4.1: Overview on the modifications for each of the datasets [1].

In addition to this qualification of the information, it is necessary to make some clarifications detailing the possible inconveniences that the use of these datasets entails, which, as already mentioned from the beginning, are scarce sets and sometimes contain repetitive or inaccurate information. Thus, one of the first problems that we find is the large number of words that these datasets have, which although at first, it may seem a positive point, even a nuance of quality, the reality is that it has a negative impact on the final results of the image captioning models, mainly due to the fact that by having a greater variety of lexicon and vocabulary for each image, a greater disparity is generated between the distribution of the training and evaluation datasets, which leads to a more difficult generation of captions.

Besides this greater disparity between the distribution of the set, it is also added to the problem that not only are there great differences in the captions but also that a smaller number of types of words represents a deficient vocabulary, directly affecting the quality of the sentence and reducing the performance of the descriptions, while also affecting the frequency of words can be related to the previous paragraph, since a large number of low-frequency words leads directly to data loss [178].

Adding to this factor is that for a wide variety of vocabulary and lexicon, the largest available data set, i.e. RSICD remains minuscule and less diversified compared to popular natural image caption datasets such as COCO-captions containing 330,000 images, or for example with Flickr8K [179] where the authors found that the latter had a vocabulary three times larger than RSICD, with only about 8,000 images. The number of repeated captions is also considered a worrying concept, since the more number of repeated captions, the more images are described equally, being a clear example the caption ”many buildings and green trees are in a dense residential area”, which is repeated more than 500 times in the data set.

	Sydney		UCM		RSICD	
Modifications	**Before**	**After**	**Before**	**After**	**Before**	**After**
Words in Training Dataset	224	196	349	318	2603	2077
Words in Validation Dataset	111	103	225	216	1168	1044
Words in Test Dataset	104	97	222	211	1562	1388
All Words	237	201	368	327	3325	2628
Word Frequency <2	31	12	36	14	1523	1029
Word Frequency <5	71	45	68	40	2070	1459
Word Frequency <10	97	65	103	67	2391	1743
Words Only in Validation Dataset	10	4	11	7	310	247
Words Only in Test Dataset	6	3	11	3	433	323

Table 4.2: Changes in the number of words before and after modifying the datasets [1].

Being a last relevant point and that was pointed out by the authors, the fact that the vocabulary used in the validation division differed significantly from that used in the training and evaluation division, which could not only generate serious problems in the results training when defining the early stop criteria, but could mean a reduction in the final score obtained by the evaluation metrics, and presenting the model as of poorer quality, thus considering future work in the area to be almost strictly necessary. which can address the mentioned limitations, either through a correction and unification of the current databases or through the proposal of a new set of reference data – see Table 4.2.

4.1.2 Evaluation Metrics

To evaluate the performance of the model, it is necessary to select the appropriate metrics to check the performance of the proposed method and, in turn, to be able to compare the results of different image captioning methods proposed by previous studies. Using as a reference some articles already published in the field [128, 180, 181], the traditional n-gram overlap metrics will be adopted: BiLingual Evaluation Substitute (BLEU) [182], Recall-Oriented Understudy for Gisting Evaluation (ROUGE)-L [183], Metric for Translation Evaluation with Explicit Ordering (METEOR) [184] and CIDEr [28].

Starting with the Bilingual Evaluation Understudy (BLEU) metric, it compares the N-gram overlap ratio between the generated sentence and the corresponding reference sentence, being a direct estimate of the n-gram precision, where $\hat{y}$ is the candidate sequence, and the ground truth sequence, s the sequence of n-grams of the candidate sequence and $c(s, y)$ represents the frequency of s in y.

$$\mathrm{p}_n = \frac{\sum_s \min(c(s,\hat{y}), c(s,y))}{\sum_s c(s,\hat{y})}, \qquad \mathrm{p}_n = \frac{\sum_{ngram \in c} \min\left(\max_{i=1,\ldots,k} \mathrm{Count}_{ri}(\mathrm{ngram}), \mathrm{Count}_c(\mathrm{ngram})\right)}{\sum_{ngram \in c} \mathrm{Count}_c(\mathrm{ngram})}$$

In this way, the final score of the BLEU metric then corresponds to a weighted geometric mean of the precision score pn, with a range of [0-1], where, the fact that the BLEU metric is a score that is only based on precision caused certain problems, which led the authors to propose a brevity penalty (BP) and be able to prevent very short generated sentences from obtaining very high scores and long sentences from obtaining a worse score performance. Being in this new calculation, concerning the calculation of the BLEU score, N represents the largest sequence of n-grams and wn is a positive weight factor, which the authors define as $N = 4$ and $wn = 1/N$ in their line of the base, where r represents the total length of the reference corpus and c represents the length of the generated sequence.

$$\mathrm{BP} = \begin{cases} 1 & \text{if } \mathrm{len}(c) \geq \mathrm{len}(r) \\ \exp(1 - \frac{len(r)}{len(c)}) & \text{otherwise} \end{cases} \qquad \mathrm{BLEU} = \mathrm{BP} \times \exp(\sum_{n=1}^{4} \lambda_n \log p_n)$$

As we have already seen, BLEU remains to this day a very common metric in the evaluation of natural language generation, where its score consists of a decimal value that varies from 0 to 1, where the scores that are most close to 1, will correspond to those with a precision closer to the reference sequence, although several weaknesses have been pointed out, mainly due to its calculation based on precision and the problems that this entails.

In a more specific case, both Reiter in his 2018 proposal [185], was argued that there was not enough evidence to support that this metric was the best one to evaluate NLP systems, and the same authors of the METEOR metric proposed an article [186] where the main problems that could result from the use of this metric as the only way of evaluating the system were pointed out, the most relevant being 4 main weaknesses. Among these four, we first find the lack of recall, this is because they were able to show through experimentation that the brevity penalty introduced to compensate for the use of long and complex sentences does not adequately compensate for this difference and the generation continued to benefit from this failure. short sentences.

Next, we have the use of higher-order n-grams which does not correctly take into account the importance of grammatically which, added to the lack of an explicit word match between the candidate and reference sequences, directly results in incorrect matches appearing and a lack of observation in grammatically. Finally, and looking at a more mathematical scope, the fact that it is a geometric mean, as well as its equations, can indirectly lead to a score of 0 as long as one or more components of the N-gram are scored 0, indicating that this metric cannot make sense at the entire sequence level.

Next, we find the METEOR metric, which evaluates the sentence generated by a score based on word-to-word matches and therefore calculates a word alignment between two sentences, being the metric for the evaluation of the translation with explicit ordering. This metric, in the same way as BLEU, fixes its final score through a decimal number that will oscillate between 0 and 1 and which will be based on the harmonic mean of the precision and the recall of the uni-gram. Being therefore the precision, the ratio between the number of coincidences of uni-grams and the total number of uni-grams in the set of hypotheses and where the recovery refers to the difference between the number of coincidences of uni-grams and the total number of uni-grams in the reference set. The following formula corresponds to the parameterized harmonic mean of P and R [187].

$$\mathrm{F}_{mean} = \frac{\mathrm{P} \cdot \mathrm{R}}{\alpha \cdot \mathrm{P} + (1 - \alpha) \cdot \mathrm{R}}$$

Therefore, the total number of fragments is directly related to the segmentation of the generated sentence and, in turn, inversely related to the degree of order, which causes that the greater the number of fragments, the greater the segmentation of the sentence and the greater the sanction and therefore achieving the lowest penalty, when the whole of the sentence is a single match, which would mean that the total number of fragments would be equal to 1. Being therefore in the equations the value of gamma, which determines the maximum penalty, value which ranges between 0 and 1 and which in turn directly affects the functional relationship between the gamma itself and the fragmentation, using as default values gamma equal to 0.5 and beta equal to 3, according to the original proposal of the authors in their article.

$$\mathrm{Pen} = \gamma \cdot \left(\frac{ch}{m}\right)^{\beta} \qquad \mathrm{Score} = (1 - \mathrm{Pen}) \cdot \mathrm{F}_{mean}$$

Following then we find the ROUGE metric, this is a retrieval oriented metric originally designed for the evaluation of automatic summarization, where it uses the F-measure based on the longest common subsequence (Longest Common Subsequence (LCS)) [188] to evaluate the similarity between the candidate sentence and the reference sentence, automatically counting LCS the longest N-gram.

Although there are several different versions of the ROUGE metric, such as ROUGE-N, where the number of matching 'n-grams' between our model-generated text and a 'reference' is measured, with the N representing the n-gram we are using, where for ROUGE-1 we would be measuring the match rate of unigrams and for ROUGE-2 and ROUGE-3 they would use bigrams and trigrams respectively. Or we also have ROUGE-S or skip-gram, which allows us to search for consecutive words from the reference text, which appear in the model output but are separated by one or more words, thus allowing us to add a degree of leniency to our matching of N-grams [189].

In this work the ROUGE-L version is used, this being perhaps one of the most popular versions, this version of the metric measures the longest common subsequence (LCS) between the output of our model and the reference. In other words, a count is made of the longest sequence of tokens that is shared between both chains, this being the longest shared sequence, which would indicate greater similarity between the two sequences, thus being able to apply the respective recovery calculations. and precision in the same way as for the other versions, but this time the match variable would be replaced by the LCS [190].

It should be noted that despite offering advantages over the other versions, such as the lack of a requirement for a predefined N-gram length in the input, the fact that the sequence is not evaluated in its entirety and that therefore, the only evaluation that is taken into account is that of the longest sequence of the text, which implies that the alternative LCS, which could contain relevant information regarding the performance of the system, are being overlooked or rather ignored.

$$\mathrm{R}_{lcs} = \frac{\mathrm{LCS}(X,Y)}{m} \qquad \mathrm{P}_{lcs} = \frac{\mathrm{LCS}(X,Y)}{n} \qquad \mathrm{F}_{lcs} = \frac{(1+\beta^2)\,\mathrm{R}_{lcs}\,\mathrm{P}_{lcs}}{\mathrm{R}_{lcs} + \beta^2\,\mathrm{P}_{lcs}}$$

Another commonly used metric originally proposed for image captioning tasks is the Consensus-Based Image Description Evaluation (CIDEr) metric, which relies on both accuracy and memory to calculate how well a The generated sentence matches the consensus of a collection of image captions, thus taking advantage of concepts such as Term-Frequency (Term frequency (TF)) and Inverse-Document-Frequency (Inverse Document Frequency (IDF)) – see Figure 4.3 [191].

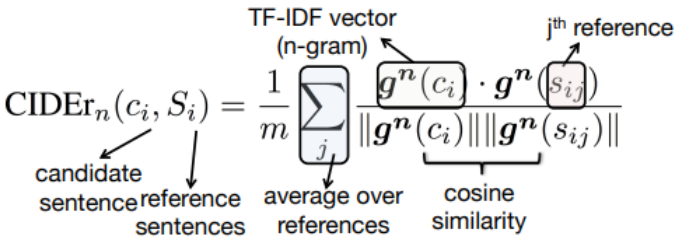


Figure 4.3: Overview of CIDErn score formula [28].

These terms act together, then being called Term Frequency – Inverse document frequency (Term frequency – Inverse document frequency (TF-IDF)) and refer to the frequency of occurrence of a term in a document or a collection of documents, thus being a metric that allows expressing numerical how relevant a word is to a document in a collection and the value of which increases proportionally to the number of times a word appears in the document while being offset by the frequency of the word in the document collection, thus allowing for the fact that some words are generally more common.

$$\mathrm{CIDEr}(c_i, S_i) = \sum_{n=1}^{N} w_n \, \mathrm{CIDEr_n}(c_i, S_i)$$

Being therefore the CIDErn score, calculated for a specific N-gram, the use of prior knowledge registered, for the calculation of the average cosine similarity and therefore the final score, corresponds to a weighted sum of each previously calculated CIDErn. Demonstrating in a recent study [185] that this is one of the metrics considered more robust since it is the one that is most correlated with human evaluations, which makes it much more "natural" than any of the automatic metrics mentioned up to now.

Following Louis and Nenkova [192], the following table depicts Kendall's tau rank correlations between automatic metrics and human assessments determined on a system-level. To prevent any quality issues connected with crowd-sourced ratings, the statistics were derived using the available expert annotations, where the length of model outputs was not controlled, and the automatic metrics were produced in a multi-reference context, using the original reference summary provided in the CNN/DailyMail dataset and 10 additional summaries from Kry'sci'nski – see Table 4.3 [193].

Metric	**Coherence**	**Consistency**	**Fluency**	**Relevance**
BLEU	0.1176	0.0735	0.3321	0.2206
METEOR	0.2353	0.6324	0.6126	0.4265
ROUGE-L	0.0735	0.1471	0.2583	0.2353
CIDER	0.1176	-0.1912	-0.0221	0.1912

Table 4.3: Kendall's tau correlation coefficients of expert annotations computed on a system-level along four quality dimensions with automatic metrics using 11 reference summaries per example.

4.2 Experimental Results

In the initial test, different versions of the hyper-parameters have been compared to find the optimal values for them. Variations were made to check the different efficiencies of values such as learning rate, momentum or weight decay, being the best optimizer for our model the SGD with a learning rate of 1e3, with a momentum of 0.9. Experiments have been made by training initially only the new layers (cross-attention and final projection layers) plus the decoder, and after that finally training the entire model until the validation loss stops decreasing, using a linear scheduler.

The learning rate schedule is a predefined framework that adjusts the learning rate as the training progresses according to a pre-defined schedule, this schedule can be very varied but the most common are time-based, step-based and exponential. In our experiments a linear schedule is used, where the learning rate decreases linearly from the initial lr set in the optimizer to 0, after a warmup period during which it increases linearly from 0 to the initial lr set in the optimizer. This is an adequate method to apply in the training process, since in the early iterations the learning rate is set to be large to reach a set of weights that are good enough and, over time, these weights are fine-tuned to reach higher accuracy by leveraging a small learning rate.

In all our experiments, the beam search decoding was also used, with a beam size of 5. Whereas greedy decoding calculates the best option based on the very next token only, the beam search algorithm checks for multiple tokens in the future and assesses the quality of all of these tokens combined. From this search, it will be returned multiple potential output sequences (considered 'beams'), and these will be evaluated individually one by one according to the input images. Taking into account their scores, the one with the best score will be chosen and this will be the one that will be used for the final evaluation of the caption generation.

Tables 4.4 - 4.6, represent the results of our model for the different datasets, even the smaller ones (UCM-captions and Sydney-captions). For the Sydney dataset, the results obtained exceed those of several different previous studies, in all or almost all the metrics. In turn, the results shown by the tables for the largest datasets, such as RSCID, present relatively low scores compared to the current state-of-the-art.

These significantly low results concerning the current state-of-art may occur for several reasons. The hypothesis deduced after analyzing in detail the differences between the datasets used, is that both the visual contents and the captions that make them up are completely different in terms of diversity and internal complexity.

To carry out an internal analysis of the captions of each dataset, an example of both Sydney-captions and RSCID is chosen and the possible differences are evaluated, which reflected the complete content of these datasets.

For Sydney-caption we find some captions:

- **Caption[0]:** A narrow runway on the river bank.
- **Caption[1]:** There is a curved narrow runway on the river bank.
- **Caption[2]:** A narrow curved runway are on the river bank.

Table 4.4: Comparison of our model and the previous state-of-the-art on Sydney-captions

Method	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDER
FC Attention + LSTM [120]	0.807	0.554	0.409	0.711	2.203
SM Attention + LSTM [120]	**0.814**	0.580	0.411	0.719	2.302
Structured attention [136]	0.779	0.586	0.395	0.729	2.379
Cross-hierarchy attention [137]	0.817	0.591	—	0.721	2.291
ML Attention + Semantic [22]	0.823	0.600	0.420	0.706	2.311
Denoising-based fusion [138]	0.832	0.585	—	0.721	3.819
Multi-level attention [1]	0.790	0.605	0.474	0.735	2.181
Full Transformer	0.790	**0.649**	**0.601**	**0.812**	**3.320**

Table 4.5: Comparison of our model and the previous state-of-the-art on UCM-captions

Method	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDER
FC Attention + LSTM [120]	0.813	0.635	0.417	0.750	2.995
SM Attention + LSTM [120]	0.815	0.645	0.424	0.763	3.186
Structured attention [136]	0.853	0.714	0.463	0.814	**3.348**
Cross-hierarchy attention [137]	0.823	0.659	—	0.756	3.19
ML Attention + Semantic [22]	0.833	0.662	0.437	0.796	3.168
Denoising-based fusion [138]	0.830	0.634	—	0.731	3.295
Multi-level attention [1]	**0.886**	**0.727**	**0.522**	**0.844**	3.307
Full Transformer	0.662	0.454	0.368	0.680	2.341

Table 4.6: Comparison of our model and the previous state-of-the-art on RSICD

Method	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDER
CSMLF [117]	0.575	0.221	0.212	—	0.500
FC Attention + LSTM [120]	0.745	0.457	0.339	0.633	2.366
SM Attention + LSTM [120]	0.757	0.461	0.351	0.645	2.356
Structured attention [136]	0.701	0.393	0.329	0.729	1.703
Cross-hierarchy attention [137]	0.770	0.471	—	0.651	2.363
ML Attention + Semantic [22]	0.759	0.462	0.354	0.699	2.361
VRTMM+SCST [30]	0.793	0.511	0.372	—	2.793
Multi-level attention [1]	**0.805**	**0.516**	**0.471**	**0.723**	**2.771**
Full Transformer	0.560	0.309	0.298	0.581	1.964

While for RSCID:

- **Caption[0]:** there are two baseball fields an athletic track four tennis courts and a swimming pool.
- **Caption[1]:** on the lawn with trees around there are two baseballfields, a swimming pool, a ground track field, four tennis court and other sports grounds.
- **Caption[2]:** two lines of green trees are near a playground next to two baseball fields and three tennis courts.

These two examples reflect the internal structure of each of the datasets and the possible reason why the program has lower performance in the second. These problems are highlighted above all in the notable differences in the sentence length, the size of the vocabulary used in the collection, the sentence length in terms of tokens, and the captions that make up the RSICD being superior in number for all these factors.

Note that the phrases that make up the RSICD also contain a great variation among them, being practically different in structure, even varying the writing of the category to which they belong, referencing as ”baseballfields” or ”baseball field”, depending on the chosen caption. All of these factors can directly affect the CLIP-text tokenizer, causing the decoder to have a very tough time representing the more complex sentences, but if it performs fairly well on the small dataset that is much simpler.

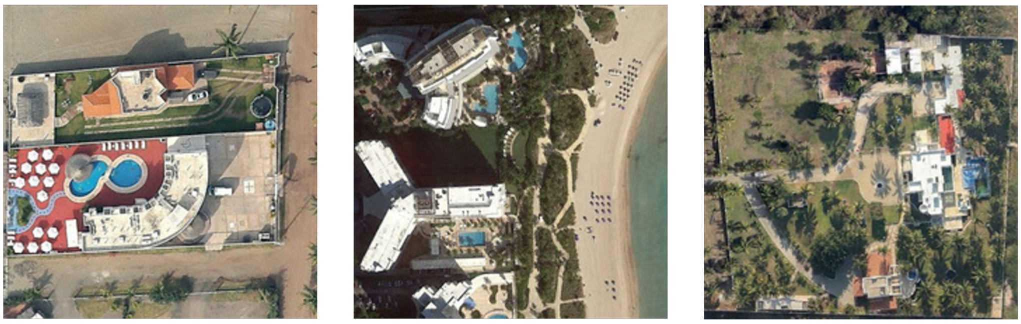


Figure 4.4: Example of different images of the RESORT type scene in RSCID.

On the other hand, according to the analysis of the visual content, we can see how Sydney-captions consist of seven classes, including residential, airport, meadow, rivers, ocean, industrial and runway and a total of 613 images, while RSCID contains a total of 10,921, where about 30 types of scenes can be differentiated. All this not only affects the number of different elements that the CLIP model must classify since it is trained to carry out classification tasks without any problem but similar to what happened in the analysis of the captions, the existing differences among the images of the same class for the RSCID are large since it has a great variability of images - see Figure 4.4.

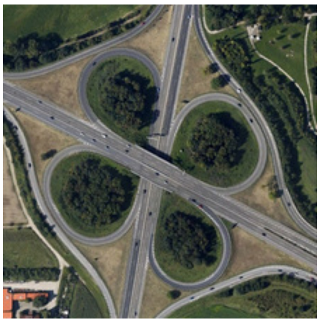
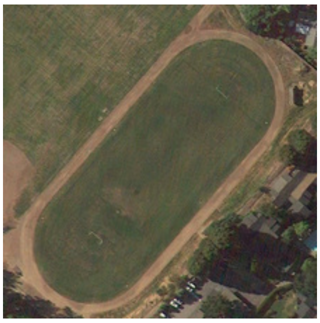
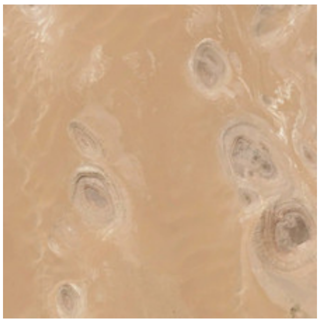
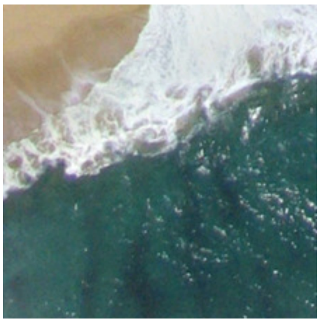
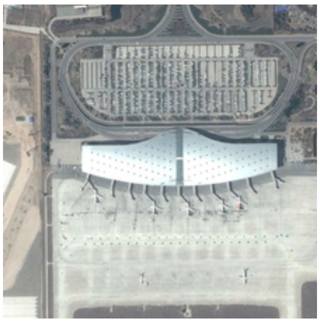
Next, in the qualitative analysis, an extensive comparison is made of the generated captions and the possible failures that may be affecting the bad result in large datasets.

Table 4.7, includes both the original input images, the real or sample caption, and the caption generated by the system. This has been done to be able to demonstrate that the metrics used to evaluate the NLP tasks are still far from being assimilated into human understanding since the generated sentences shown are almost identical with variations of the original, where the result at the level of semantics is identical, but when evaluated they have obtained average grades, which are far from the reality.

Carrying out a more complete analysis, we examine the different examples, since each one presents a different evaluation problem. Starting with the VIADUCT image, the caption ”Many green trees and several buildings” is identical both in terms of position within the sentence and word composition. However, it is internally inverted, thus affecting the evaluation result. Similarly, this also happens with the PLAYGROUND example, where the caption is identical, but the sentence complement is at the beginning or the end, because a sentence is written in the active voice and another in the passive. On the other hand, the DESERT caption has a different order and synonyms such as bare land are used, but the meaning is completely the same. This also happens in the OCEAN example, where only the position adjective changes, possibly being interchangeable for ocean and beach. Finally, we find the AIRPORT example where due to the characteristics of the aerial view, the system confuses an airport with a factory, being perfectly understandable even for a human.

The examples are intended to clarify that the small errors made by the system have probably been due mainly to limitations in the evaluation methods concerning semantics and the inclusion of synonyms.

Table 4.7: Examples of generated captions on RSICD

Images	Captions
	**Real Caption:** Many green trees and several buildings are near a viaduct . **Predicted Caption:** Several buildings and many green plants are near a viaduct .
	**Real Caption:** A playground is near some buildings and green trees . **Predicted Caption:** Many buildings and some green trees are near a playground .
	**Real Caption:** The large land is a vast desert . **Predicted Caption:** There is a bare land in the desert .
	**Real Caption:** In front of the ocean is a vast beach . **Predicted Caption:** Near the beach is a vast ocean .
	**Real Caption:** There are a lot of bare land around the airport . **Predicted Caption:** There are a lot of bare land around the factory .

Chapter 5

Conclusions and Future Work

Throughout this document, a new method for captioning remote sensing images has been presented. This model is based on the structure of a full Transformer-based encoder-decoder where both the encoder and the decoder use components from a fine-tuned version of the CLIP model from the RSICD dataset.

In this last chapter, the learning produced in the consideration of the proposed premises are demonstrated, reflection is made on the relevance of the problem established in the argument, considerations are provided regarding the appropriate way of thinking about the problem, all this framed in a brief summary of the main points addressed in the work, where the results obtained are exposed and analyzed and the most important findings are highlighted, as well as the possible directions for future work.

5.1 Conclusions

The specific characteristics that this new structure provides are explored in depth, as well as the peculiarities that make it unique and that allow the adjustment of both the encoder and the decoder, since when using the model pre-entering CLIP which are entering to perform a wide variety of tasks, which can be leveraged through natural language prompts to enable zero-trigger transfer to many existing datasets, enabling competitive performance compared to task-specific supervised models, although there is still a large room for improvement.

The experimental results presented reveal that our method is effective, surpassing the current state of the art in most of the metrics, only for the small datasets, such as the Sydney-captions database. For larger datasets, such as RSICD, results are still significantly below the current state-of-the-art.

The integration of large pre-trained encoder-decoder architectures must be taken into account. In our case the full Transformer is an effective, simple and promising means, in the tasks of natural language processing and computer vision to improve the performance and results of the captioning task. At the same way, it is necessary to take into account attention mechanisms, especially in the captioning processes of remote sensing images and in general of NLP, since they are the basis of the generation of an understandable text, with fluency and cohesion.

5.2 Future Work

The results of our study are promising and with great future development, especially in regard to its application to large volumes of data, since due to the characteristics of the created system and the particularities of these images, there is a wide margin for improvement, study and development.

Some of the improvements that could be taken into account for future studies and thus create a model capable of also competing in its use with massive datasets, would consist of the introduction of an auxiliary language model that takes advantage of the captions of similar images and that therefore, it will facilitate the generation of a new captions through previously collected information.

Another of the possible improvements is the implementation of pre-processing to the input images using CLIP, in a similar way to what is done in the study called ClipCap, where it is use the CLIP encoding as a prefix to the caption, by employing a simple mapping network, and then fine-tunes a language model to generate the image captions.

Finally, it is also important to reinforce the idea that both the existing datasets and the metrics used for evaluation are of low semantic diversity, mainly due to the lack of investment in the field of remote sensing image captioning and also that it is complicated to execute an analysis and evaluation of such a complex task in a simple way. However, a great development is expected in the future, especially focused on the creation of better datasets, always taking into account the possible current limitations, but with a great progression since they have a great margin for improvement.

Bibliography

[1] Y. Li, S. Fang, L. Jiao, R. Liu, and R. Shang, “A multi-level attention model for remote sensing image captions,” *Remote. Sens.*, vol. 12, p. 939, 2020.

[2] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[3] A. Oppermann, “Artificial intelligence vs. machine learning vs. deep learning.” https://towardsdatascience.com/artificial-intelligence-vs-machine-learning-vs-deep-learning-2210ba8cc4a 2019.

[4] S. Mukherjee, “Activation function neural networks fundamentals for deep learning.” https://www.postinweb.com/activation-function-neural-networks/, 2021.

[5] J. Feng, X. He, Q. Teng, C. Ren, H. Chen, and Y. Li, “Reconstruction of porous media from extremely limited information using conditional generative adversarial networks,” *Physical Review E*, vol. 100, 09 2019.

[6] J. J. Moolayil, “A layman’s guide to deep neural networks,” *Towards Data Science*, 2019.

[7] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, “Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting,” *Water*, vol. 12, no. 5, 2020.

[8] M. Phi, “Illustrated guide to lstm’s and gru’s: A step by step explanation.” https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb8 2018.

[9] Y. Gavrilova, “A guide to deep learning and neural networks.” https://serokell.io/blog/deep-learning-and-neural-network-guide, 2020.

[10] S. Saha, “A comprehensive guide to convolutional neural networks.” https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd 2018.

[11] M. Deshpande, “Introduction to convolutional neural networks for vision tasks.” https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/, 2017.

[12] B. Pembelajaran, “Convolutional neural networks (cnn) introduction.” https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/.

[13] S. JOGLEKAR, “Residual neural networks as ensembles.” https://codesachin.wordpress.com/tag/residual-networks/, 2017.

[14] S.-H. Tsang, "Densenet — dense convolutional network (image classification)." https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803, 2018.

[15] Maxime, "What is a transformer?." https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04, 2019.

[16] T. M. Learners, "Transformer: la tecnología que domina el mundo." https://www.themachinelearners.com/transformer/.

[17] G. Boesch, "Vision transformers (vit) in image recognition." https://viso.ai/deep-learning/vision-transformer-vit/, 2020.

[18] K. Goutham, "Image captioning." https://medium.com/@kanukollugouthamviswatej/image-captioning-ffa555d5ba3b, 2020.

[19] P. Radhakrishnan, "Image captioning in deep learning." https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2, 2017.

[20] Katnoria, "Caption this! image caption using neural networks." https://www.katnoria.com/nic-p1/, 2019.

[21] S. Sarkar, "Image captioning using attention mechanism." https://medium.com/swlh/image-captioning-using-attention-mechanism-f3d7fc96eb0e, 2020.

[22] Z. Yuan, X. Li, and Q. Wang, "Exploring multi-level attention and semantic relationship for remote sensing image captioning," *IEEE Access*, vol. PP, pp. 1–1, 12 2019.

[23] M. Murali, "Applications of remote sensing and geographic information system (gis) in archaeology," 01 2015.

[24] S. Nikiforova, T. Deoskar, D. Paperno, and Y. Winter, "Geo-aware image caption generation," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3143–3156, 2020.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[26] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "Cptr: Full transformer network for image captioning," *ArXiv*, vol. abs/2101.10804, 2021.

[27] Y. Gao, J. Shi, J. Li, and R. Wang, "Remote sensing scene classification based on high-order graph convolutional network," *European Journal of Remote Sensing*, vol. 54, pp. 1–15, 01 2021.

[28] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015.

[29] X. Wang, Y. Wu, Y. Ming, and H. Lv, "Remote sensing imagery super resolution based on adaptive multi-scale feature fusion network," *Sensors (Basel, Switzerland)*, vol. 20, 2020.

[30] X. Shen, B. Liu, Y. Zhou, J. Zhao, and M. Liu, "Remote sensing image captioning via variational autoencoder and reinforcement learning," *Knowl. Based Syst.*, vol. 203, p. 105920, 2020.

[31] Y. Wang, J. Xu, Y. Sun, and B. He, "Image captioning based on deep learning methods: a survey," *ArXiv*, 2019.

[32] Z. Zhang, W. Diao, W. Zhang, M. Yan, X. Gao, and X. Sun, “Lam: Remote sensing image captioning with label-attention mechanism,” *Remote. Sens.*, vol. 11, p. 2349, 2019.

[33] O. Simeone, “A very brief introduction to machine learning with applications to communication systems,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[34] G. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, pp. 1527–54, 08 2006.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[36] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–44, 05 2015.

[37] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, 2013.

[38] R. Sarikaya, G. E. Hinton, and A. Deoras, “Application of deep belief networks for natural language understanding,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.

[39] A.-r. Mohamed, G. E. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[40] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-regulated splicing code,” *Bioinformatics*, vol. 30, pp. i121–i129, 06 2014.

[41] B. Alipanahi, A. Delong, M. Weirauch, and B. Frey, “Predicting the sequence specificities of dna- and rna-binding proteins by deep learning,” *Nature biotechnology*, vol. 33, 07 2015.

[42] S. Zhang, J. Zhou, H. Hu, H. Gong, L. Chen, C. Cheng, and J. Zeng, “A deep learning framework for modeling structural features of RNA-binding protein targets,” *Nucleic Acids Research*, vol. 44, pp. e32–e32, 10 2015.

[43] J. Smolander, M. Dehmer, and F. Emmert-Streib, “Comparing deep belief networks with support vector machines for classifying gene expression data from complex disorders,” *FEBS Open Bio*, vol. 9, no. 7, pp. 1232–1248, 2019.

[44] T. P. Carvalho, F. Soares, R. Vita, R. da Piedade Francisco, J. P. Basto, and S. G. S. Alcalá, “A systematic literature review of machine learning methods applied to predictive maintenance,” *Comput. Ind. Eng.*, vol. 137, 2019.

[45] S. A. Oke, “A literature review on artificial intelligence,” *International journal of information and management sciences*, vol. 19, pp. 535–570, 2008.

[46] A. Petrillo, M. Travaglioni, F. D. Felice, R. Cioffi, and G. Piscitelli, “Artificial intelligence and machine learning applications in smart production: Progress, trends and directions,” 2019.

[47] J.-P. Haton, “A brief introduction to artificial intelligence,” *IFAC Proceedings Volumes*, vol. 39, no. 4, pp. 8–16, 2006. 9th IFAC Symposium on Automated Systems Based on Human Skill and Knowledge.

[48] Y. Gavrilova, “A guide to deep learning and neural networks.” https://serokell.io/blog/deep-learning-and-neural-network-guide, 2020.

[49] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An introductory review of deep learning for prediction models with big data,” *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[50] J. TORRES.AI, “Learning process of a deep neural network: How do artificial neural networks learn?,” *Towards Data Science*, 2020.

[51] J. H. et M.C, *L’intelligence artificielle*. PUF, 3 ed., 1993.

[52] J.-P. Haton, “A brief introduction to artificial intelligence,” *IFAC Proceedings Volumes*, vol. 39, no. 4, pp. 8–16, 2006. 9th IFAC Symposium on Automated Systems Based on Human Skill and Knowledge.

[53] Y. Bengio, “Learning deep architectures for ai,” *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[54] B. Y. Goodfellow I. and C. A., “Deep learning,” 2016.

[55] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[56] A. Graves, “Generating sequences with recurrent neural networks,” *ArXiv*, vol. abs/1308.0850, 2013.

[57] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[58] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” *Neural Computation*, vol. 12, pp. 2451–2471, 10 2000.

[59] R. Cahuantzi, X. Chen, and S. Güttel, “A comparison of lstm and gru networks for learning symbolic sequences,” *ArXiv*, 2021.

[60] S. Bansari, “Introduction to how cnns work.” https://medium.datadriveninvestor.com/introduction-to-how-cnns-work-77e0e4cde99b, 2019.

[61] C. C. Chatterjee, “Basics of the classic cnn: How a classic cnn (convolutional neural network) work?.” https://medium.datadriveninvestor.com/introduction-to-how-cnns-work-77e0e4cde99b, 2019.

[62] N. Sharma, V. Jain, and A. Mishra, “An analysis of convolutional neural networks for image classification,” *Procedia computer science*, vol. 132, pp. 377–384, 2018.

[63] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[64] V. A. Gurumurthy, “Encoder-decoder based cnn and fully connected crfs for remote sensed image segmentation,” *ArXiv*, 2019.

[65] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, “Deep learning classification of land cover and crop types using remote sensing data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[66] R. Kemker, C. Salvaggio, and C. Kanan, “Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 60–77, 2018. Deep Learning RS Data.

[67] P. H. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in *ECCV*, 2016.

[68] G. Lin, A. Milan, C. Shen, and I. Reid, “Refinenet: Multi-path refinement networks for high-resolution semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1925–1934, 2017.

[69] J. Brownlee, “How do convolutional layers work in deep learning neural networks?,” *Machine Learning Mastery*, 2019.

[70] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An introductory review of deep learning for prediction models with big data,” *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[71] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ArXiv*, 2014.

[72] G. Boesch, “Vgg very deep convolutional networks (vggnet).” `https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/`.

[73] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, and K. Keutzer, “Imagenet training in minutes,” in *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1–10, 2018.

[74] 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)*, pp. 770–778, 2016.

[75] D. Lin, S. Talathi, and S. Annapureddy, “Fixed point quantization of deep convolutional networks,” in *International conference on machine learning*, pp. 2849–2858, PMLR, 2016.

[76] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[77] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks,” in *NIPS*, 2015.

[78] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[79] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European conference on computer vision*, pp. 630–645, Springer, 2016.

[80] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[81] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

[82] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[83] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *ICML*, 2013.

[84] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *AAAI*, 2017.

[85] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *ArXiv*, vol. abs/1502.03167, 2015.

[86] B. Raj, “A simple guide to the versions of the inception network,” *Towards Data Science*, 2018.

[87] 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.

[88] U. Ankit, “Transformer neural network: Step-by-step breakdown of the beast.” `https://towardsdatascience.com/transformer-neural-network-step-by-step-breakdown-of-the-beast-b3e096dc`, 2020.

[89] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ArXiv*, 2020.

[90] X. Chen, C.-J. Hsieh, and B. Gong, “When vision transformers outperform resnets without pre-training or strong data augmentations,” *ArXiv*, 2021.

[91] S. Paul and P.-Y. Chen, “Vision transformers are robust learners,” *ArXiv*, vol. 2, no. 3, 2021.

[92] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[93] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, “How to train your vit? data, augmentation, and regularization in vision transformers,” *ArXiv*, 2021.

[94] L. Srinivasan and D. Sreekanthan, “Image captioning-a deep learning approach,” 2018.

[95] S. MARTIN, “Startup aids visually impaired with guided service powered by ai.” `https://blogs.nvidia.com/blog/2019/03/01/startup-aids-visually-impaired-with-guided-service-powered-by-ai/`, 2019.

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[97] 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*, pp. 248–255, 2009.

[98] “Imagenet website.” `https://www.image-net.org/index.php/`.

[99] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

[100] Z. Yang, Y.-J. Zhang, Y. Huang, *et al.*, “Image captioning with object detection and localization,” in *International Conference on Image and Graphics*, pp. 109–118, Springer, 2017.

[101] J. Aneja, A. Deshpande, and A. G. Schwing, “Convolutional image captioning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5561–5570, 2018.

[102] J.-Y. Pan, H.-J. Yang, P. Duygulu, and C. Faloutsos, “Automatic image captioning,” 07 2004.

[103] 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*, pp. 3156–3164, 2015.

[104] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, pp. 2048–2057, PMLR, 2015.

[105] A. Roy, “A guide to image captioning.” `https://towardsdatascience.com/a-guide-to-image-captioning-e9fd5517f350`, 2020.

[106] K. Doshi, “Image captions with deep learning: State-of-the-art architectures.” `https://towardsdatascience.com/image-captions-with-deep-learning-state-of-the-art-architectures-329057`, 2021.

[107] A. Chowdhry, “Image caption generator: Cnn-lstm architecture and image captioning.” `https://blog.clairvoyantsoft.com/image-caption-generator-535b8e9a66ac`, 2021.

[108] D. Malyk, “Exploring deep learning image captioning.” `https://mobidev.biz/blog/exploring-deep-learning-image-captioning`, 2021.

[109] K. Doshi, “Image captions with deep learning: State-of-the-art architectures.” `https://towardsdatascience.com/image-captions-with-deep-learning-state-of-the-art-architectures-329057`, 2021.

[110] I. I. Amal, D. H. Widyantoro, and A. Umam, “Mobilenet-based neural image caption model in title generation for product’s images,” in *2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA)*, pp. 1–6, 2020.

[111] T. Ahmed, A. Kapadia, M. Swaminathan, and V. Potluri, “Up to a limit? privacy concerns of bystanders and their willingness to share additional information with visually impaired users of assistive technologies,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, 09 2018.

[112] M. R. Oleksii Sidorov, Ronghang Hu and A. Singh, “Textcaps: a dataset for image captioning,” *Facebook AI Research*, 2021.

[113] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, “Deeplogo: Hitting logo recognition with the deep neural network hammer,” *ArXiv*, 2015.

[114] S. C. Kumar, M. Hemalatha, S. B. Narayan, and P. Nandhini, “Region driven remote sensing image captioning,” *Procedia Computer Science*, vol. 165, pp. 32–40, 2019. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.

[115] X. Shen, B. Liu, Y. Zhou, and J. Zhao, “Remote sensing image caption generation via transformer and reinforcement learning,” *Multimedia Tools Appl.*, vol. 79, p. 26661–26682, sep 2020.

[116] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4511–4523, 2013.

[117] B. Wang, X. Lu, X. Zheng, and X. Li, "Semantic descriptions of high-resolution remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 8, pp. 1274–1278, 2019.

[118] B. Wang, X. Zheng, B. Qu, and X. Lu, "Retrieval topic recurrent memory network for remote sensing image captioning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 256–270, 2020.

[119] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2183–2195, 2017.

[120] X. Zhang, X. Wang, X. Tang, H. Zhou, and C. Li, "Description generation for remote sensing images using attribute attention mechanism," *Remote. Sens.*, vol. 11, p. 612, 2019.

[121] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *GIS '10*, 2010.

[122] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2175–2184, 2015.

[123] Z. Yuan, X. Li, and Q. Wang, "Exploring multi-level attention and semantic relationship for remote sensing image captioning," *IEEE Access*, vol. 8, pp. 2608–2620, 2020.

[124] Z. Shi and Z. Zou, "Can a machine generate humanlike language descriptions for a remote sensing image?," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3623–3634, 2017.

[125] S. Wu, X. Zhang, X. Wang, C. Li, and L. Jiao, "Scene attention mechanism for remote sensing image caption generation," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2020.

[126] F. Hu, G.-S. Xia, W. Yang, and L. Zhang, "Recent advances and opportunities in scene classification of aerial images with deep models," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4371–4374, 2018.

[127] B. Qu, X. Li, D. Tao, and X. Lu, "Deep semantic understanding of high resolution remote sensing image," in *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1–5, 2016.

[128] X. Lu, B. Wang, X. Zheng, and X. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, pp. 2183–2195, 2018.

[129] Y. Wang, H. Ma, K. Alifu, and Y. Lv, "Remote sensing image description based on word embedding and end-to-end deep learning," *Scientific Reports*, vol. 11, 2021.

[130] G. Sumbul, S. Nayak, and B. Demir, "Sd-rsic: Summarization-driven deep remote sensing image captioning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6922–6934, 2020.

[131] X. Lu, B. Wang, and X. Zheng, “Sound active attention framework for remote sensing image captioning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 3, pp. 1985–2000, 2019.

[132] A. Aker and R. Gaizauskas, “Summary generation for toponym-referenced images using object type language models,” *Proceedings of the International Conference RANLP-2009*, pp. 6–11, 01 2009.

[133] W. Cui, X. He, M. Yao, Z. Wang, J. Li, Y. Hao, W. Wu, H. Zhao, X. Chen, and W. hong Cui, “Landslide image captioning method based on semantic gate and bi-temporal lstm,” *ISPRS Int. J. Geo Inf.*, vol. 9, p. 194, 2020.

[134] X. Fan, A. Aker, M. Tomko, P. Smart, M. Sanderson, and R. Gaizauskas, “Automatic image captioning from the web for gps photographs,” 03 2010.

[135] X. Zhang, Q. Wang, S. Chen, and X. Li, “Multi-scale cropping mechanism for remote sensing image captioning,” in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 10039–10042, 2019.

[136] R. Zhao, Z. Shi, and Z. Zou, “High-resolution remote sensing image captioning based on structured attention,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[137] C. Wang, Z. Jiang, and Y. Yuan, “Instance-aware remote sensing image captioning with cross-hierarchy attention,” in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 980–983, 2020.

[138] W. Huang, Q. Wang, and X. Li, “Denoising-based multiscale feature fusion for remote sensing image captioning,” *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 3, pp. 436–440, 2021.

[139] R. Ramos and B. Martins, “Remote sensing image captioning with continuous output neural models,” in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL ’21, (New York, NY, USA), p. 29–32, Association for Computing Machinery, 2021.

[140] H. Larochelle, D. Erhan, and Y. Bengio, “Zero-data learning of new tasks,” in *AAAI*, 2008.

[141] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, 2009.

[142] J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, “Predicting deep zero-shot convolutional neural networks using textual descriptions,” *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4247–4255, 2015.

[143] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, “Zero-shot learning through cross-modal transfer,” in *NIPS*, 2013.

[144] A. Krizhevsky, “Learning multiple layers of features from tiny images,” *University of Toronto*, 05 2012.

[145] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, “Devise: A deep visual-semantic embedding model,” in *NIPS*, 2013.

[146] A. Li, A. Jabri, A. Joulin, and L. van der Maaten, “Learning visual n-grams from web data,” *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4193–4202, 2017.

[147] K. Desai and J. Johnson, “Virtex: Learning visual representations from textual annotations,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11157–11168, 2021.

[148] M. B. Sariyildiz, J. Perez, and D. Larlus, “Learning visual representations with caption annotations,” in *ECCV*, 2020.

[149] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. Langlotz, “Contrastive learning of medical visual representations from paired images and text,” *ArXiv*, vol. abs/2010.00747, 2020.

[150] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” 2019.

[151] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. J. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” *ArXiv*, vol. abs/2005.14165, 2020.

[152] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, “Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[153] S. F. Dodge and L. Karam, “A study and comparison of human and deep learning recognition performance under visual distortions,” *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, 2017.

[154] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, and W. Brendel, “Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness,” *ArXiv*, vol. abs/1811.12231, 2019.

[155] M. A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. M. Nguyen, “Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4849, 2019.

[156] A. Borji, “Objectnet dataset: Reanalysis and correction,” *ArXiv*, vol. abs/2004.02042, 2020.

[157] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. L. Zhu, S. Parajuli, M. Guo, D. X. Song, J. Steinhardt, and J. Gilmer, “The many faces of robustness: A critical analysis of out-of-distribution generalization,” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329, 2021.

[158] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, “Learning robust global representations by penalizing local predictive power,” in *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

[159] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–567, 2019.

[160] R. Zhang, “Making convolutional networks shift-invariant again,” *ArXiv*, vol. abs/1904.11486, 2019.

[161] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” *ArXiv*, vol. abs/1508.07909, 2016.

[162] R. Mokady, A. Hertz, and A. H. Bermano, “Clipcap: Clip prefix for image captioning,” *ArXiv*, vol. abs/2111.09734, 2021.

[163] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. abs/2101.00190, 2021.

[164] R. Socher and L. Fei-Fei, “Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora,” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 966–973, 2010.

[165] H. Agrawal, K. Desai, Y. Wang, X. Chen, R. Jain, M. Johnson, D. Batra, D. Parikh, S. Lee, and P. Anderson, “nocaps: novel object captioning at scale,” *International Conference on Computer Vision*, pp. 8947–8956, 2019.

[166] 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 *European conference on computer vision*, pp. 740–755, Springer, 2014.

[167] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” *ArXiv*, vol. abs/1504.00325, 2015.

[168] N. M. Shazeer and M. Stern, “Adafactor: Adaptive learning rates with sublinear memory cost,” *ArXiv*, vol. abs/1804.04235, 2018.

[169] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.

[170] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, “Marian: Fast neural machine translation in C++,” in *Proceedings of ACL 2018, System Demonstrations*, (Melbourne, Australia), pp. 116–121, Association for Computational Linguistics, July 2018.

[171] J. Tiedemann and S. Thottingal, “OPUS-MT — Building open translation services for the World,” in *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, (Lisbon, Portugal), 2020.

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

[173] Y. Yang and S. Newsam, “Uc merced land use dataset.” `http://weegee.vision.ucmerced.edu/datasets/landuse.html`, 2010.

[174] N. G. PROGRAM, “Usgs national map urban area imagery collection.” `https://www.usgs.gov/programs/national-geospatial-program/national-map`.

[175] M. Hodosh, P. Young, and J. Hockenmaier, “Framing image description as a ranking task: Data, models and evaluation metrics (extended abstract),” *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, 2013.

[176] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[177] X. Li, X. Zhang, W. Huang, and Q. Wang, “Truncation cross entropy loss for remote sensing image captioning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 6, pp. 5246–5257, 2021.

[178] R. P. Ramos and B. Martins, “Using neural encoder-decoder models with continuous outputs for remote sensing image captioning,” *IEEE Access*, 2022.

[179] F. Community, “Flickr 8k.” `https://www.kaggle.com/adityajn105/flickr8k`.

[180] X. Zhang, X. Wang, X. Tang, H. Zhou, and C. Li, “Description generation for remote sensing images using attribute attention mechanism,” *Remote Sensing*, vol. 11, p. 612, mar 2019.

[181] B. Wang, X. Lu, X. Zheng, and X. Li, “Semantic descriptions of high-resolution remote sensing images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, pp. 1274–1278, 2019.

[182] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *ACL*, 2002.

[183] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *ACL 2004*, 2004.

[184] A. Lavie and A. Agarwal, “Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments,” in *WMT@ACL*, 2007.

[185] E. Reiter, “A structured review of the validity of bleu,” *Computational Linguistics*, vol. Just Accepted, pp. 1–8, 2018.

[186] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in *IEEvaluation@ACL*, 2005.

[187] A. Celikyilmaz, E. Clark, and J. Gao, “Evaluation of text generation: A survey,” *ArXiv*, vol. abs/2006.14799, 2020.

[188] R. A. Wagner and M. J. Fischer, “The string-to-string correction problem,” *Journal of the ACM (JACM)*, vol. 21, no. 1, pp. 168–173, 1974.

[189] J. Briggs, “The ultimate performance metric in nlp.” `https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460`, 2021.

[190] S. Ravichandir, “Understanding rouge-l,” in *Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT*, p. 352, Packt Publishing, 2021.

[191] K. S. Jones, “A statistical interpretation of term specificity and its application in retrieval,” *J. Documentation*, vol. 60, pp. 493–502, 2004.

[192] A. Louis and A. Nenkova, “Automatically assessing machine summary content without a gold standard,” *Computational Linguistics*, vol. 39, pp. 267–300, 2013.

[193] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, “Evaluating the factual consistency of abstractive text summarization,” *ArXiv*, vol. abs/1910.12840, 2020.