Understanding Vision-Language Models

Alejandro Hernández Díaz

Master of Science in Computer Vision, Robotics and Machine Learning

from the

University of Surrey

Department of Electrical and Electronic Engineering

Faculty of Engineering and Physical Sciences

University of Surrey

Guildford, Surrey, GU2 7XH, UK

September 2022

Supervised by: Miroslaw Bober

DECLARATION OF ORIGINALITY

I confirm that the project dissertation I am submitting is entirely my own work and that any material used from other sources has been clearly identified and properly acknowledged and referenced. In submitting this final version of my report to the JISC anti-plagiarism software resource, I confirm that my work does not contravene the university regulations on plagiarism as described in the Student Handbook. In so doing I also acknowledge that I may be held to account for any particular instances of uncited work detected by the JISC anti-plagiarism software, or as may be found by the project examiner or project organiser. I also understand that if an allegation of plagiarism is upheld via an Academic Misconduct Hearing, then I may forfeit any credit for this module or a more severe penalty may be agreed.

Understanding Vision-Language models

Alejandro Hernández Díaz

Author Signature  Date: 06/09/2022

Supervisor's name: Miroslaw Bober

WORD COUNT

Number of Pages: 67

Number of Words: 15092

ABSTRACT

The ongoing data revolution combined with the latest advances in technology, stimulated the application of deep learning to many data analysis tasks. Bigger and more complex architectures are now designed to also concurrently process multiple data modalities. One of the most prominent fusions in multi-modal machine learning is the one between Visual and Linguistic data.

Following the ongoing trend in both Computer Vision and Natural Language Processing fields, pre-trained Transformers have also become the de-facto approach for their fusion. However, there are numerous other design choices present in the development of VL architectures and scientist are yet to find an optimal combination of them.

In order to accelerate research, this dissertation investigates how different design choices affect the performance and behaviour of VL models. For this purpose, a representative subset of 5 SoTA VL architectures is selected, analysed, and benchmarked in two distinct tasks: Image Classification and Text to Image Retrieval.

The principal observations drawn are as follows. In terms of visual inputs, models using the more computationally expensive Faster R-CNN features are outperformed by architectures which utilise non-overlapping flattened image patches, in both reported metrics and inference/training time. Further, ViT-based architectures exhibit superior visual-processing capabilities compared to BERT-based models. Additionally, it is substantially harder for BERT-based networks to achieve competitive visual understanding via VL pre-training tasks, than it is for ViT-based approaches to obtain the linguistic processing capabilities they lack. Finally, in the case of learning-related design choices, Masked Multi-Modal Learning and Multi-Modal Alignment Prediction were the only tasks found to be essential for the correct functioning of the selected models in the two benchmarks, while additional objectives, such as the regression of masked visual features or VqA, were found to offer a substantial performance boost.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

This project focuses on the topic of multi-modal artificial learning, which is concerned with the ability of machine learning algorithms to concurrently process data from different modalities (Image and video, text, audio...). To be more specific, the two modalities which will be studied are Vision and Language.

There are many approaches to cross-processing visual and language data and, scientists are yet to discover an optimal approach to it among all the available options. In this work a representative subset of current state-of-the-art Vision-Language architectures will be analysed and benchmarked on common tasks, in order to not only gain a better understanding about how their specific components allow for a successful cross-modality fusion, but also determine how each individual combination of design choices influence their ability to extract useful joint representations.

1.1 Background and Context

The field of Artificial Intelligence has achieved outstanding advances in the last decade. The computational power granted by cutting edge GPU (graphical processing unit) technology and the collection of web data into large-scale datasets, have allowed the creation and development of Deep Learning, revolutionising our society.

These Deep-based approaches, such as GPT3 [10] or BERT [15] in NLP and Efficientnet [57] or R-CNN [21] in Computer Vision, are obtaining formidable results in their respective fields. However, the outside world is inherently multi-modal in terms of stimuli so, in order to make further advances towards more complex real applications, these existing architectures have to be adapted to concurrently processing data from different modalities.

The fusion of these two previously mentioned fields i.e. Computer vision and Natural Language Processing, is one of the most prominent in artificial multi-modal learning. This popularity comes predominantly as a consequence of the following three factors: to begin with, sight is proven to be deeply correlated with language in human learning. Secondly, it is thought that language-aligned visual representations can improve the current performance in purely-visual

tasks. And finally, the availability of abundant large-scale datasets and benchmarks for both modalities facilitates the process tremendously.

Following their great success in the fields of vision and language alone, the use of pre-trained Transformer variants has also become the de-facto choice for their integration, achieving state-of-the-art performance on every VL task available. Nevertheless, there are numerous other design choices that play a crucial role when it comes to developing a VL architecture and, researchers are yet to find the combination which provides optimal cross-modal capabilities.

In order to accelerate the evolution of vision-language research, this dissertation intends to shed some light on the development of SoTA architectures, by carrying out an appropriate comparison of a representative subset of available methods within the field, thoroughly analysing how different combinations of design choices affect each model's performance.

1.2 Objectives

The main objectives of this project are the following.

- Identify the most prominent tasks in the current integration of vision and language research, as well as the available datasets used for each of them.
- Review existing state-of-the-art architectures proposed for solving the previously introduced tasks and select a subset of representative models for their assessment.
- Set up and run various tests in order to benchmark said models on appropriate tasks, datasets and metrics.
- Extensively discuss the obtained results, providing a thorough description of the key model-specific mechanisms and design choices that influence their performance.
- Summarize the entire experimentation process, as well as the relevant literature, in a clear and well-structured report.

1.3 Achievements

The principal contributions present in this piece of work are as follows:

- A thorough analysis of 5 state-of-the-art Vision-Language architectures is provided. Regardless of the high level of detail, the explanations are intended to be easy to follow, highlighting the most important contributions that each publication supplied to the field.
- The subset of selected VL models are applied to a purely visual task, i.e. Image classification, which has not been done before in the literature. Furthermore, they are also benchmarked on category based image retrieval on a tremendously complex image dataset. The implementation details for each of the experiments is also provided so the reader can reproduce the results if desired.
- The results obtained as part of the experimentation process are rigorously discussed, concluding with the identification of key design choices which both positively and negatively impact VL performance. Furthermore, ideas on how to extend this piece of work are also provided.
- Additionally, the image classification performance of the selected VL models is compared to the achieved by 4 non-linguistic architectures i.e. CNNs, in order to provide information towards affirming the conjecture which states that language-aligned visual representations are beneficial for purely visual tasks.

1.4 Overview of Dissertation

The contents of this dissertation are structured as summarised bellow:

Chapter 1. This first chapter provides an introduction to the work carried out in this project. It also contains the objectives and most notable achievements that will be presented in the upcoming sections.

Chapter 2. The second chapter presents a thorough description of the background knowledge necessary to adequately understand this investigation. To begin with, the methods within the individual fields of Computer Vision and Natural Language Processing are covered. This makes sure that a non-expert reader gets an understanding of what the different artificial intelligence subfields are concerned with. Subsequently, the importance of learning useful data representations in machine learning is introduced. Finally, the process of merging these uni-modal representa-

tions in VL research is discussed. Due to the high-importance of transformers in this work, their architecture was also covered in this section.

Chapter 3. The first technical chapter covers all the relevant methodology utilized for the purpose of this investigation. It starts by revisiting the research question in order to justify the upcoming experiments. Since the ultimate goal is to better understand the impact that different model-specific mechanisms have in their final performance, the next subsection analyses each of the selected architectures to the necessary level of detail for an appropriate discussion of the results. Finally, the chapter also presents the dataset and downstream tasks selected for the benchmarking.

Chapter 4. The second technical chapter covers the experimental set up of the two selected benchmarks, in order to allow for a reproduction of the results if desired. Everything ranging from hyperparameter choice, to model adaptation strategies is included.

Chapter 5. The last technical chapter provides the desired analysis of the impact that different design choice combinations have in the model's cross-modal capabilities. For each benchmark, the results are presented and the architectures are covered in ascending order of performance. At the end of each subsection, the main conclusions are drawn.

Chapter 6. The final chapter of this dissertation summarizes the work carried out to date, boiling down the main conclusions and evaluating the approach taken during its completion. Finally, some directions and ideas to further extend this investigation are also included.

2 BACKGROUND THEORY AND LITERATURE REVIEW

The following subsections of the report summarise the theoretical concepts necessary to understand the work carried out in this project. To begin with, prior to diving into multi-modal representation research the individual fields of computer vision and natural language processing will be discussed. Subsequently, the attention will drift from only solving specific problems to also learning useful data representations in the process. Thereafter, the current trends in the fusion of said uni-modal representations in Vision-Language research will be discussed.

2.1 Computer Vision: an Overview of Methods

Computer vision (CV) is the area of artificial intelligence (AI) that works towards enabling computers to extract meaningful information from various types of visual input i.e. digital images or videos. The systems can subsequently utilize this information to take actions or make informed decisions [46].

It is an extensive field of active research yet far from being solved and, due to its vast complexity, it is often divided into specific subtasks in order to facilitate its study. Some examples of computer vision tasks are *image classification*, *image segmentation* or *image retrieval*.

Traditional computer vision adopted a hierarchical approach to solving these kinds of problems, where algorithms that detected edges, curves and corners were used as the building blocks towards a high-level understanding of visual data [41].

This approach required of expert engineers to hand-craft useful features for each specific problem/domain, in order to transform the raw image pixels into a more suitable **representation** to work with. In computer vision, said features are measurable characteristics which can accurately describe the content of an image, and were often based on structural or chromatic information. This low-level information can be further processed to create more robust image descriptors including Histograms of Oriented Gradients (HOG) [13] or SURF features [9].

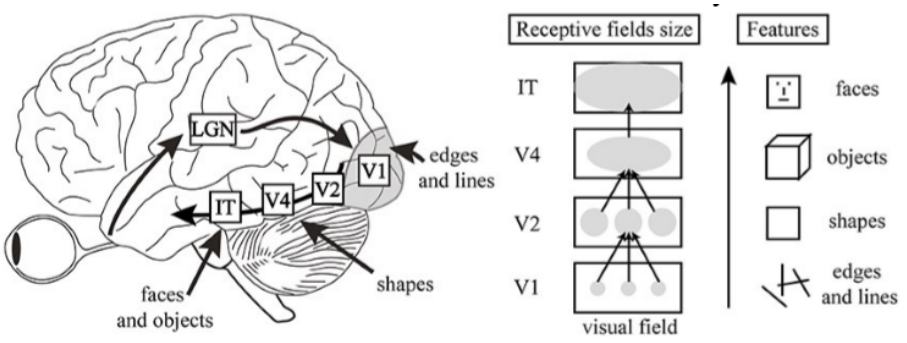

Figure 2.1: Hierarchical approach to Computer Vision. Low-level features are combined to obtain a more complex understanding of visual inputs. [2]

Said descriptors were often paired with classifiers from a distinct subfield of Artificial Intelligence: Machine learning, which focuses on the design of systems that can adapt to perform a desired task, not by following hard coded instructions, but by analysing patterns in the examples used to train them.

Recent advances in the field of machine learning caused a change of paradigm in computer vision. Since 2012, the majority of CV tasks have been dominated by a subset of ML algorithms that comprise the so called sub-field of Deep Learning.

Deep learning research is concerned with creating algorithms that ”mimic” the human’s brain behaviour called Artificial Neural Networks (ANN), which are multi layered computing structures composed of individual processing units called neurons. These new algorithms replaced the traditional pipeline, as they now could be trained completely end to end, performing **feature extraction** and decision-making in an automated manner.

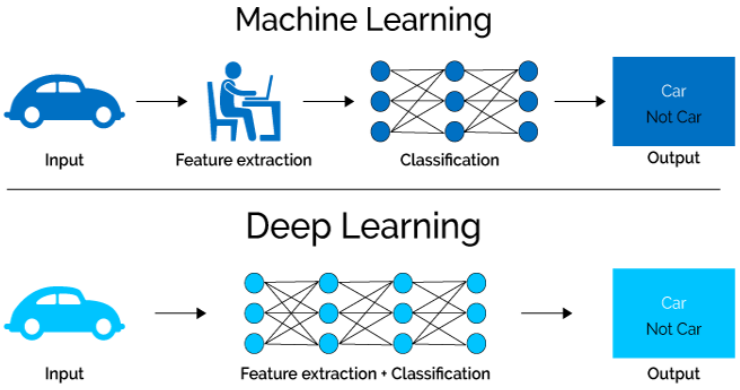

Figure 2.2: Traditional vs Modern/Deep Computer Vision pipeline [2].

In the CV field, most available solutions are based on an ANN-variant called Convolutional Neural network (CNN), whose most salient characteristic is that it ”replaces” the previously introduced neurons with learnable filters which are convolved with the input image in order to extract valuable information. The first modern CNN (LeNet [33]) was introduced by Yann LeCun 1998, but it was not until the creation of AlexNet [32], that from 2012 onwards, these deeper algorithms became remarkably popular in the vision community.

Since then, different variants of the CNN architecture such as VGG [54], ResNet [22] or the R-CNN (Region-Based CNN) series [21] [51] among others, were considered state-of-the-art in many CV tasks.

It was not until 2020 that the researchers' focus started drifting away from convolution as the scales tipped towards attention-based approaches. With the creation of ViT [17], the transformer architecture (neural network variant introduced in the latter section), was adapted and applied to visual inputs, surpassing the previously mentioned algorithms and establishing a new state of the art that continues to this date.

2.2 Natural language processing: an Overview of Methods

Natural language Processing (NLP) is a multi disciplinary field which combines computer science, machine learning and linguistics in order to allow artificial systems to synthesize, understand and generate human language.

Human-language processing is an extremely broad topic thus, following the same trend as computer vision, its research is divided in solving various subtasks within it. Some of the most common examples of NLP tasks are *text classification*, *question answering* or *sentiment analysis*

In the early days, scientists took a statistical approach to solving some of these tasks. However, traditional machine learning techniques were introduced to the topic, transforming the workflow into a similar pipeline as the one present in the CV field [19].

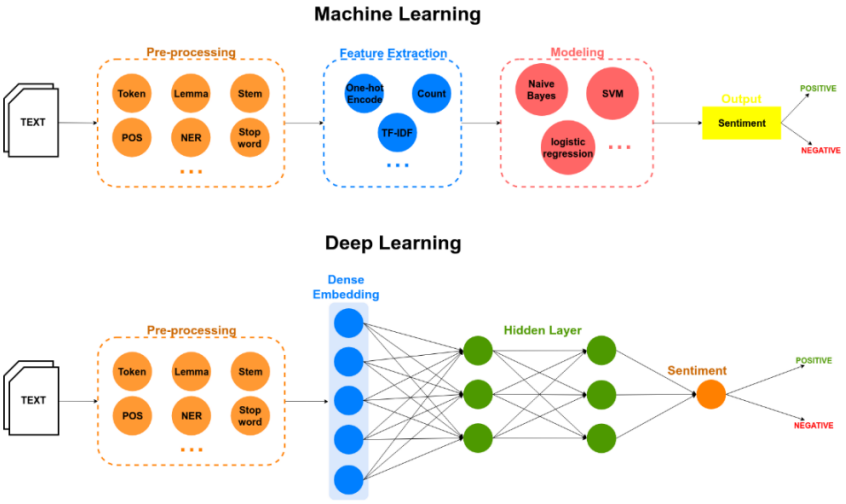

Figure 2.3: Traditional vs Modern/Deep Natural Language Processing pipeline [14].

In the early 2010s, this traditional approach was left behind with the introduction of deep learning models. This new viewpoint allowed for the processing of sequence data (i.e. language) itself in a more direct manner, instead of being constrained to purely statistics.

However, ANNs are not able to process text data (words) directly so the first step in this paradigm transition was to find a suitable **representation** for it. This resulted in the creation of word embeddings. These embeddings are learned vector representations of words (or subwords) that encode semantic meaning, allowing for similar words to have similar values in the vector space. Some of the most popular techniques used to derive these embeddings are the Continuous Bag of Words and Continuous Skip-Gram Model, which were combined to create the so called Word2Vec architecture [43].

Text is inherently *sequential* (i.e. the ordering of words matter when it comes to understanding text) hence the existing deep learning architectures had to be adapted to this characteristic. The first ANN-based architecture specialized for sequential data modelling was the so called *Recurrent Neural Network* (RNN) [53]. RNNs process each input consecutively so that the ordering is considered, while also working in a *autorregresive* manner, which allows for the context to be taken into account when processing each individual word. However, RNNs are not able to retain said context information for long, so the creation of Long-Short term memory (LSTM) [24] cells and Gated recurrent Units (GRU) [12] was necessary in order to grant better memory capabilities to the systems.

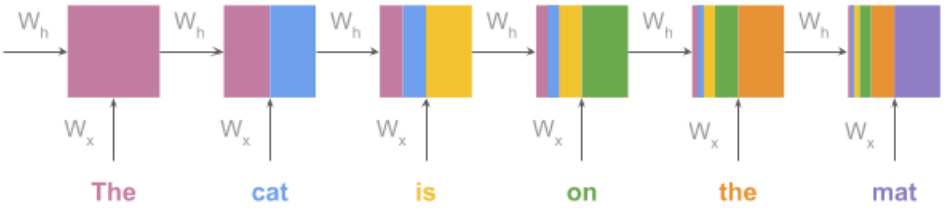

Figure 2.4: As the input sequence grows, the information retained by the model concerning earlier tokens starts to vanish [27].

The previously introduced models obtained state-of-the-art performance in many language tasks for some time. Moreover, these results would further improve with the application of attention techniques, which would revolutionize this and many other fields within AI.

In 2017, a novel encoder-decoder architecture denominated the *Transformer* [58] outperformed its competitors in various NLP tasks, and was later found to be extremely versatile, as it also achieved state-of-the-art results in other unrelated fields. Said architecture goes back to the parallel processing of inputs which was present in ANNs and is based solely on attention mechanisms . This allows the model to dynamically ”attend” to different parts of the input sentence in order to introduce more context. **The Transformer architecture is a fundamental part of this dissertation and is thoroughly discussed in the context of Vision-Language in a latter subsection (2.4.1).**

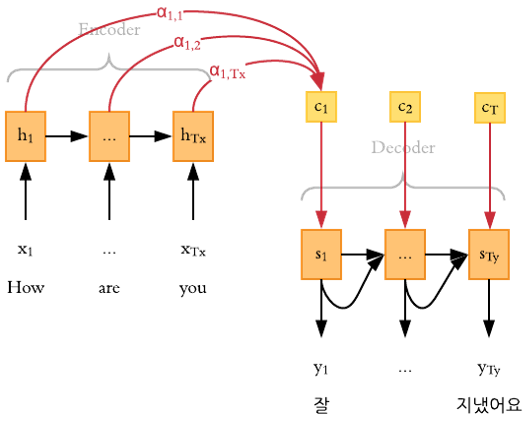

Figure 2.5: Illustration of the attention mechanism as presented in early encoder-decoder architectures. The decoder is now able to dynamically change its context according to the importance of each input token at every time-step [5].

Several modifications of the original Transformer model, such as *BERT* [15] (which discards the decoder part, building an encoder-only architecture) or the *GPT series* [49] [50] [10] (which do the counterpart, focusing only on the decoder stream of the transformer) are currently leading the field and, according to the latest research, further upgrades will allow them to continue doing so in the foreseeable future.

2.3 Learning useful data representations in Computer Vision and Natural Language Processing

The idea of data representations was briefly introduced in the previous sections when discussing visual features (in the case of CV) and word embeddings (in the NLP field). As stated there, the raw version of the data (pixel values or words, among others) is not always the most suitable form for a machine learning model. This is why, in the case of traditional ML approaches, experts were tasked with extracting useful features from entire datasets. These representations tend to be relatively compact vectors which encapsulate the most important information of the input.

With the rise of Deep Learning, researchers realised that, instead of manually deciding which representation was more suitable for each task, the network architectures could automatically learn them according to the problem being solved. This was already presented in figure 2.2, which shows how deep models typically work as automatic feature extractors connected to a final classification layer.

In the case of computer vision, when a CNN is being trained on a classification task, people tend to think that the model only learns to correctly distinguish between classes. However, the architecture is also learning to properly extract the most useful information out of it into a compact and semantically rich representation, which is fed into the final layer that performs the labelling. For a human, these final vectors do not have any apparent meaning, but upon further inspection of the dataset in this new representation space, it can be seen how the position of every data point encapsulates said information. For instance, related images would have similar representations thus, they would cluster together.

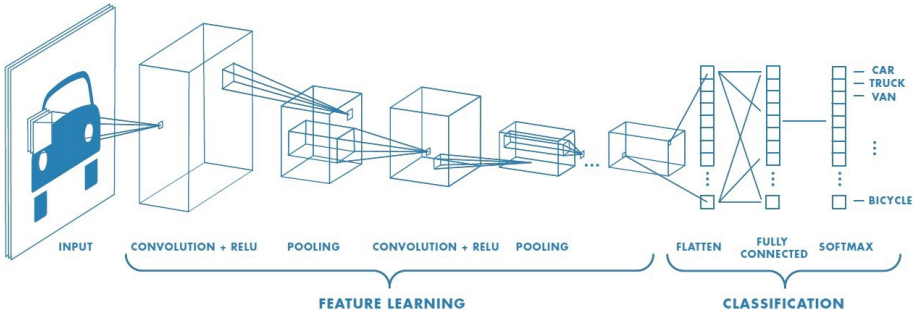

Figure 2.6: CNN architecture as a feature extractor, which transforms the representation of any input image [48].

In the NLP field, researchers focus on obtaining useful word embeddings that can be generally applied to solve any language task. Network-based approaches such as *Word2Vec* or *BERT* (among others) have been able to produce meaningful vector representations of tokens which also formed an organized space. In this case, each datapoint's location encapsulates semantic and context meaning, as well as the relationship between different tokens. The figure bellow exhibits this behaviour.

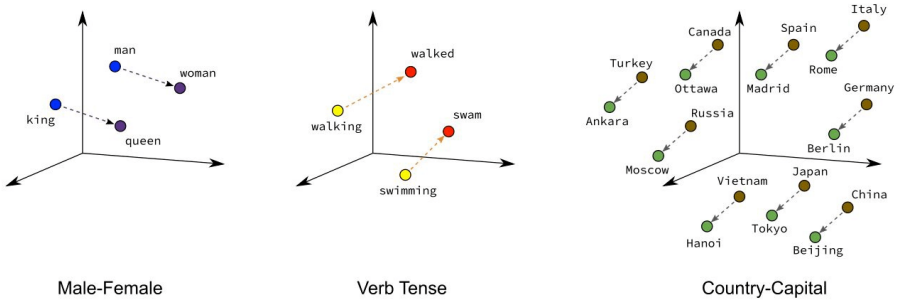

Figure 2.7: Example of an organized word embedding space, after being projected to 3 dimensions [43].

This has allowed for the trend of *Transfer Learning* to emerge, where an architecture is trained on a general task to gain the ability of extracting useful representations, to then apply this knowledge to some other downstream problems. Said behaviour mimics the way humans learn general concepts from broad tasks, which can then be repurposed for related situations if needed.

In light of the above, we now drift the attention from only focusing on accurately solving specific tasks, to also obtaining useful and reusable representations in the process.

2.4 Merging unimodal representations: Vision and Language Literature

The main difference among AI subfields is the data modality they are concerned with. CV algorithms are optimised for the processing of visual data, whereas NLP models possess specific mechanisms for a better understanding of language relationships and words.

In spite of the impressive results some of these models have obtained in their respective fields, real world stimuli usually involves multiple modalities, hence, in order for ML algorithms to adequately understand real scenarios, they need to be able to process multimodal data.

Attempting to concurrently process the *visual* and *linguistic* modalities has been one of the main focus of the research community, as it is proven that they are highly correlated in human learning.

Said fusion is also thought to have other advantages, specially from a ***Representation Learning*** perspective: The latent space generated by networks which only process visual data tends to be more *entangled* than word embedding spaces. This is mainly because entities from different categories may have similar visual features, regardless of being well differentiated linguistically. For instance, cats and dogs share key characteristics in their appearance, making them sit close in the space created by a CNN. However, the embeddings of said words would more closely resemble their relationship, as it takes their meaning into account. This is why researchers believe that language-aligned visual representations would produce more organised latent spaces that could also benefit purely visual tasks.

This modality union results in the creation of various new tasks such as:

- **Visual question answering (VQA):** Refers to the process of providing an answer to a question about a given visual input.
- **Image Captioning:** Task concerned with the generation of a linguistically well-formed description of a given visual input.
- **Text-based Visual Retrieval:** Consists on the selection of suitable images/videos from a collection, such that they match a description or query given in natural language.

This generated the question of how to successfully design algorithms which are able to jointly process both modalities. The architectural development of multi-modal VL systems is mainly focused in three distinct areas:

Visual processing. Area concerned with the extraction of *image representations* suitable for their use in VL tasks.

Some of the earliest work carried out in this area was in the context of keyword-based image retrieval by Mezaris et al. [42], which used *low-level hand crafted features* describing the color, position, dimensions and shape of entities in the images as its optical data. With the rise of deep learning in the computer vision field, VL algorithms dispensed with hand crafted representations and started to make use of pre-trained CNNs, such as VGG, as their visual processors. Said trend started with the work of Vinyals et al. [59] in the task of image captioning, but rapidly spread throughout the field [3] [20]. Furthermore, the development of visual detector networks such as the RCNN and Faster-RCNN [51] introduced the idea of extracting Region of Interest features at an object level, allowing for more flexibility than the standard grid convolutional descriptors [6]. However, such visual processing, while effective, it also was computationally expensive, hence scientists had to search for more lightweight alternatives. Due to this, some of the most recent VL architectures in the literature [35], [30] have drifted away from detector networks in favour of Vision Transformers (ViT), which present themselves as a faster alternative with almost no detriment in performance.

Linguistic processing. Area which focuses on the extraction of *Linguistic representations* suitable for their use in VL tasks.

In this case, Recurrent Neural Networks supposed the de-facto choice of language encoder [59] in the VL field. Subsequent upgrades to this architecture, such as LSTM models [4], [28], [6], took over said role until the development of BERT [15]. From 2017, the rise of the transformer architecture supposed a paradigm shift in language encoding, with state of the art publications such as Lu et al. [40] or Su et al. [55] among others.

Modality fusion. Area which centres its attention on the successful *fusion of both modalities*.

Early fusion mechanisms were constituted of simple concatenation or element-wise product of features, as presented in [29] and [3]. Further work was carried out in this area, with the development of more intricate mathematical methods such as bilinear fusion, presented by Fukui et al. [18].

However, According to recent publications, several variants of the *Transformer architecture* have also proven to be particularly suitable for this task. Due to said effectiveness throughout the entire spectrum of VL research, scientists have focused their attention on them, which now constitute the majority of available VL approaches in the literature.

Prior to further discussing Transformers in the vision and language modality integration, it is worth getting a better understanding of the inner workings of the architecture. The following subsection will provide a thorough explanation for reader to better understand the work later introduced in this report.

2.4.1 The Transformer Architecture

The *Transformer* is an encoder-decoder, autoregressive architecture introduced in 2017 for the task of machine translation. It overcame the main drawbacks of other language models (RNNs, LSTMs and GRUs) by discarding recurrence in favour of attention mechanisms. This granted a substantial speed-up in training, mitigated the long-term dependency problem and allowed parallelisation, making them the de-facto choice for NLP applications.

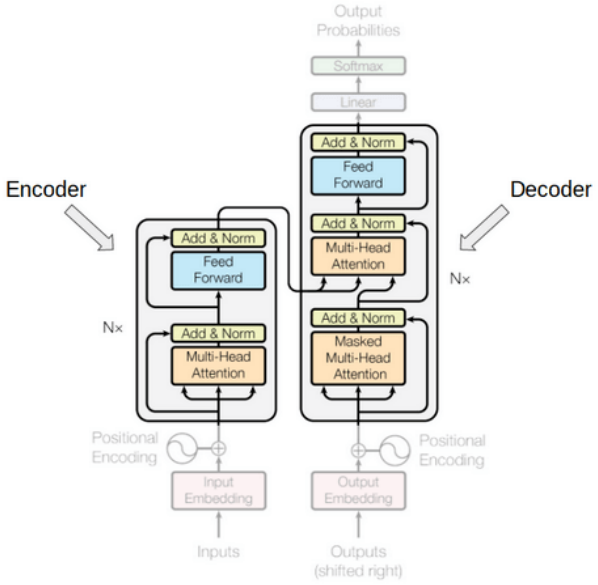

Figure 2.8: Depiction of the Transformer architecture as presented in its original paper [58].

In essence, the transformer works in a similar fashion as the RNN-based sequence-to-sequence models available at the time. The encoder stack is responsible for processing the input sequence into a suitable vector representation for the decoder to use as context when generating the desired output. According to its autoregressive nature, the transformer's decoder also makes use of the previously predicted tokens when inferring the next.

The innovation behind Transformers boils down to two main concepts:

- *Positional encodings*.
- *Encoder-decoder and self attention mechanisms*.

Positional Encodings As it was already introduced in section 2.2, text is inherently sequential. This means that, to allow for an adequate understanding of natural language, the ordering of words needs to be considered.

Even though she did not win the award, she was satisfied.

Even though she did win the award, she was not satisfied.

RNN-based models achieved this behaviour by processing their inputs consecutively. Transformers on the other hand, completely discarded this idea so, as the words of an input sentence simultaneously flow through its layers, the model has no sense of their order. To overcome this issue, a piece of information is added to the input words' embeddings for the model to interpret as their position. This information is what the authors called *positional encoding*.

More formally, given an input sentence $A = [A_0, A_1, ..., A_{N-1}]$ of length N, the *positional encoding* is a tensor P of the same shape that describes the location of each element in the sequence. When combined with A, it forms a positional-aware embedding of the input tokens.

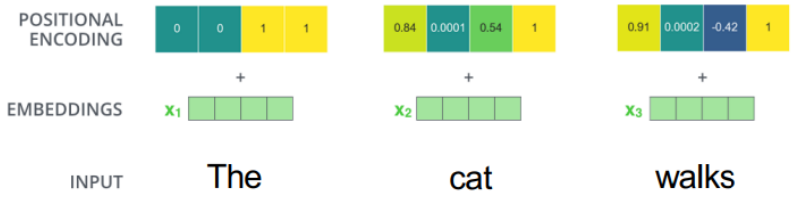

Figure 2.9: Example of positional encodings [45].

The originally proposed method was to use sinusoidal functions of varying frequencies to encode position. The following equations describe how to calculate each element i (*sine* and *cosine* for *even* and *odd* elements respectively) of a positional encoding vector PE for a certain position in the sequence pos, and dimensionality of the final embedding d_{model} [58].

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/d_{model}}) \end{aligned} \tag{2.1}$$

However, this is not the only approach to encoding this kind of information. BERT, a variant of the Transformer architecture from which the majority of VL models are derived, optimize its positional encodings as part of the learning process in conjunction to the rest of its parameters. Regardless of the computational overhead said method supposes, it allows for the model to find the optimal encoding strategy on its own, resulting in a higher flexibility [15].

Attention mechanism in Transformers The *Attention mechanism* 2.5 was introduced in the context of encoder-decoder-based machine translation, as a way of allowing the model to automatically search for parts of a source sentence that are relevant to predicting a target word, opposing to the idea of having a fixed context throughout the decoding process [8]. This idea was adapted and enhanced in the development of the Transformer architecture.

Figure 2.10: Example of attention in transformers. Darker colours represent higher attention scores i.e. closely related tokens [16].

Said mechanism is applied in three different settings throughout the architecture:

- **Encoder's self-attention:** The encoder is able to generate contextualized representations of each token by allowing them to "attend" to other relevant positions in the sequence. This permits the production of different embeddings for the same word depending on how it is used in the sentence.
- **Decoder's self-attention:** The previously predicted tokens of the target sentence are also allowed to "attend" to their neighbours for context, further enhancing the prediction process.
- **Encoder-decoder attention:** Upgrading the trend introduced by RNN-based architectures, the context used by the decoder is calculated by letting each predicted token in the target sequence "attend" to related words in the input sentence.

*- But, how does this **"attention mechanism"** work?*

The attention function can be compared to the mapping operation of a query and a set of key-value pairs to an output, in an information retrieval setting. When submitting a query, it is compared to all the database keys in order to output the values which better relate to it.

In the case of transformers, the output i.e., the encoded context-aware word representation, is a weighted combination of itself and the rest of the tokens according to the strength of their relation. The figure bellow depicts this process.

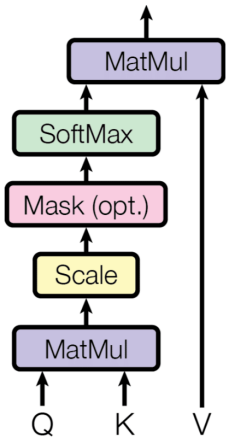

Figure 2.11: Scaled dot-product attention as presented in the transformer original paper [58].

Mathematically, the process is as follows: The input words' embeddings are linearly projected to three vector representations each, the query Q, key K, and value V, by three distinct linear layers W_Q, W_K and W_V respectively. Subsequently, for word A, the **dot-product** between its query vector and every word's key (including itself) is calculated and scaled down by the square root of d_k (length of the key vectors) to avoid large values. This works as a **similarity measure** between tokens which assigns larger values to those that closely relate. A masking step is carried out only in the decoder side to make sure that the model can only attend to already predicted positions. The results of said operations are passed through a *softmax* function, which normalizes them into the final attention weights. The final representation of word A is equal to the sum of all the value vectors multiplied by their respective weight. This process is then repeated for every input token.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.2}$$

In order to further improve this approach, the transformer divides the Q, K and V vectors, feeding each fragment into different *"heads"* that run the previously described attention mechanism in parallel, subsequently concatenating their outputs. This *"multi-headed"* setting allows the model to jointly attend to multiple positions of the input in different ways e.g. focusing on long vs short term dependencies.

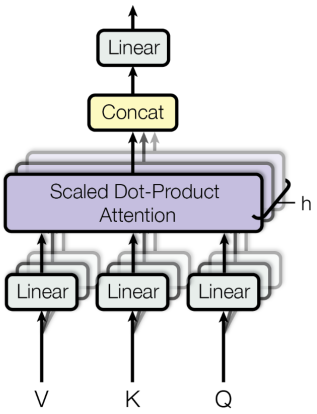

Figure 2.12: Multi-headed attention as presented in the transformer original paper [58].

2.4.1.1 Assembling the pieces together: an architectural overview

As it was already introduced in section 2.4.1, the original transformer architecture is comprised of the following two parts:

On the one hand, the ***Encoder*** stream is responsible for extracting a contextualized vector representation of the input sequence. It consists of a stack of N identical layers, where each of them is subsequently composed of two sublayers.

- A multi-headed attention layer, as described in section 2.4.1.
- A fully-connected feed-forward network, consisting of two linear layers with a Rectified Linear Unit (ReLU) activation function in between.

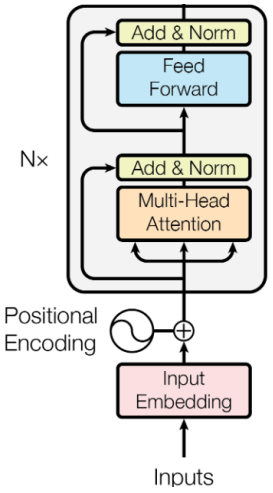

Figure 2.13: Encoder stack of X layers [58].

On the other hand, the ***Decoder*** stream utilizes the processed representation extracted by the encoder, in conjunction to the already predicted words to generate the final outputs. It consists of a stack of M identical layers, where each of them is subsequently composed of three sublayers.

- A masked multi-headed attention layer.
- A multi-headed encoder-decoder attention layer. In this newly introduced component, the query vectors come from the decoder's tokens, while the keys and values belong to the encoder's final representation.
- A fully-connected feed-forward network, consisting of two linear layers with a Rectified Linear Unit (ReLU) activation function in between.

The reason behind the masking operation performed in the decoding process is the following: during training, the decoder is fed with the desired output sentence, instead of the predicted tokens from the previous time step, as part of a strategy named "Teacher Forcing". However, in order to avoid it being conditioned by future tokens, their attention weights are masked in a way that allows each word to only attend to the ones preceding it. During inference, the decoder runs in an autoregressive manner, eliminating the need of said masks.

The decoder layers are stacked in a similar manner as the encoder stream. At the end, a final linear layer with X neurons, equal to the size of the vocabulary used, and a Softmax activation function outputs the next word's probability distribution.

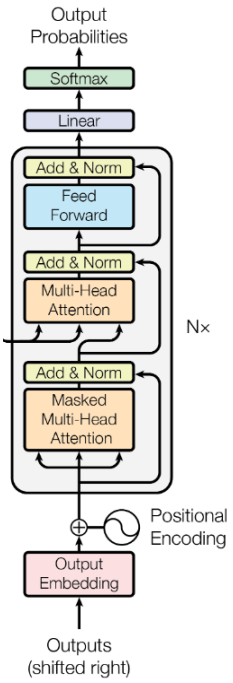

Figure 2.14: Decoder stack of X layers [58].

2.4.2 Transformers & Vision and Language: An Overview of the State of the Art

Transformers have revolutionised a multitude of subfields within AI due to their impressive ability of generating useful and reusable representations. After achieving state-of-the-art results in both stand-alone Computer Vision and NLP tasks, they were chosen as the de-facto architecture for the fusion of said modalities. Concretely, since 2019 SoTA Vision-Language (VL) models are generally based on two Transformer variants: the *BERT* model or the *Vision Transformer* (ViT) [34]. Even though the majority of implementations available opt for an encoder-only approach, some models include traditional decoders in their design for generative tasks such as *Image captioning* or *scene description* [36].

Transfer learning techniques also play a crucial role, as Transformers have proven to highly benefit from general (VL) pre-training, showing promising results in a variety of benchmarks [38].

Said architectures are mostly implemented under a common workflow: Pre-Train the network on text-image pairs while learning a general joint representation of both modalities, to subsequently leverage this knowledge when fine-tuning it on any selected downstream task [44].

The most popular pre-training objectives in the VL field are the following:

- **Masked multi-modal (VL) Learning:** pre-training task in which randomly chosen input tokens (image or text) are either masked or replaced. The model's duty is to correctly infer said tokens based on the remaining inputs.
- **Multi-modal (VL) alignment prediction:** One of the most commonly implemented pre-training objectives when attempting to align vision and language representations. Given an image and text (caption/description) pair, the model has to predict whether they are correctly matched or vice-versa.
- **Vision-Language Contrastive Learning:** Similarly to the aforementioned task, this one is concerned with the correct matching of text-image pairs. However, the usage of metric-learning objective functions, such as contrastive or triplet loss, allows for a better alignment of the vision and language representations. Said functions encourage the model to squeeze together the representation of positive V-L pairs in the vector space, while pushing away the negatives.

Said pre-training objectives (and their variants) tend to be applied using a common subset of large-scale datasets. Some of the most widely used in this stage according to the literature are summarized in table 2.1.

Table 2.1: Overview of the datasets most commonly used during pre-training.

Dataset	**# Images**	**# Image-text pairs**
SBU (2011)	875K	875K
MSCOCO (2014)	113K	567K
Flickr30k (2014)	29K	145K
Visual Genome (2017)	108K	5.4M
Conceptual Captions (2018)	3M	3M

On the other hand, regarding SoTA approaches to Vision and Language, there is a common pipeline from which most available architectures are derived from. Said *general pipeline* of VL pre-trained models is depicted in the figure bellow.

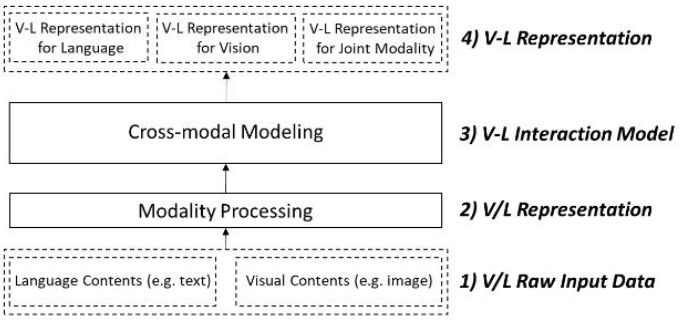

Figure 2.15: General structure of a Vision-Language Pre-Trained Model (VLPM) [38].

The pipeline described by figure 2.15 commences with raw input data (text and images/videos) being fed into *unimodal* processing modules. These modules transform the input into a suitable representation for the VL model e.g. from words to semantically rich **embeddings** and from raw pixels to **spatial-aware RoI** (Region of Interest) features. In the case of BERT based architectures, a popular choice of visual pre-processing module is the already mentioned Faster R-CNN detector network. However, as the process of extracting these features is computationally expensive, the researchers' attention is drifting in favour of ViT-based approaches, with a much lighter visual pre-processing (**flattened non-overlapping image patches**) and comparable performance (or better in some cases)[30] [35]. Transformer-based architectures which still make use of standard CNNs as their visual encoders, such as the presented by Huang et al. [25], are rapidly losing popularity.

Once the model has obtained these appropriate representations from the input data, it is ready to perform the modality fusion. According to the merging mechanism they implement, VLPMs can be classified in two groups:

- **Dual-stream models:** This modelling choice uses different encoders for the vision and language data respectively, which are kept separated. In order to allow for cross-modal interaction, dual-stream models make use of co-attention modules, where the language embeddings are able to "pay attention" to the visual features and vice-versa. ViLBERT [40], LXMERT [56] and ALBEF [35] are some examples of SoTA VL models which implement this kind of processing.

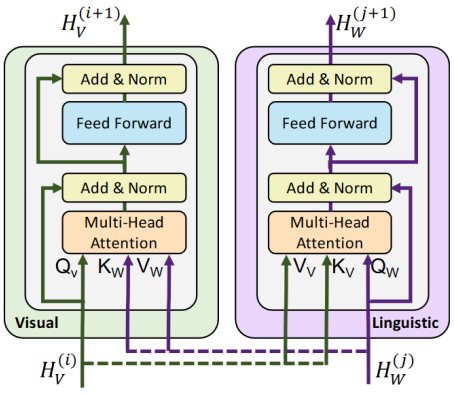

Figure 2.16: Depiction of the a type of co-attention mechanism present in dual-stream models (visual stream on the left and linguistic on the right) [40].

- **Single-stream models:** These architectures process the data by first concatenating the features extracted from both modalities, and subsequently feeding them into a single transformer encoder block. The attention mechanism from the transformer's encoder modules allow for the joint representation modelling. VL-BERT [55], OSCAR [36] and ViLT [30] (among others) constitute this category.

The following figure illustrates the previously introduced variants.

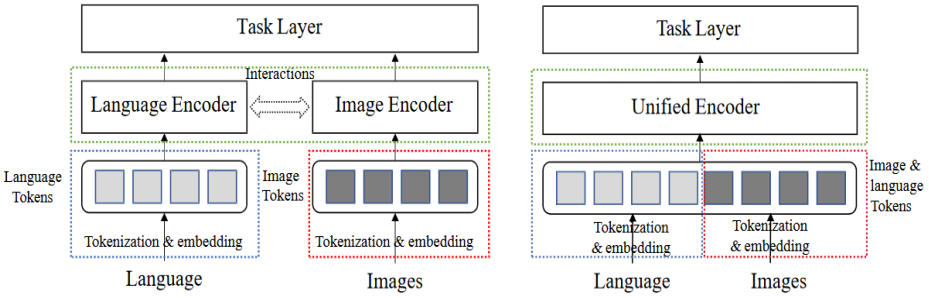

Figure 2.17: Comparison between single (left) and double (right) stream architectures. The interactions depicted in the left architecture represent the co-attention mechanism [34].

Upon completion of this cross-modal processing, the model outputs the final representation of the inputs. This representation is now suitable for the use in a variety of downstream tasks such as the previously discussed in section 2.4.This whole process has proven to be extremely helpful when tackling any VL problem. However, this is an area of active research far from being solved, and due to the subfield's novelty and the difficulty that optimally understanding such architectures

suppose, scientists are yet to agree on a clear direction towards which steer their efforts among all the available approaches.

2.5 Summary

The fields of Computer vision and Natural Language processing are currently some of the most popular inside AI as a consequence of their rapid development. This is greatly influenced by the advances achieved by machine learning research, which have allowed for a paradigm change from traditional to deep approaches.

The application of deep neural networks to said areas have eased the workload that scientist previously had since they function in a completely end to end manner. Said architectures are able to automatically extract the most salient information from raw data, forming semantically rich representations that can later be re-used in a variety of tasks.

In spite of the impressive results that unimodal architectures are obtaining, the real world is inherently multi-modal in terms of stimuli. This is why researchers are now focusing on adapting the existing architectures to fuse and process multi-modal data.

The fusion of Computer vision and Natural Language Processing has been the most popular among researchers due to their proven correlation in human learning, as well as the benefits that language-aligned visual representations would provide to purely visual tasks.

Pre-trained transformers have proven to be the most appropriate approach when tackling VL tasks, since their attention mechanism is extremely well suited to successfully perform the modality fusion. However, there are numerous design choices when it comes to deriving an architecture of this sort and, an optimal combination of them is yet to be discovered.

3 TECHNICAL CHAPTER: METHODOLOGY AND DATA

In this first technical chapter the research question is briefly revisited in order to justify the experiments conducted. Subsequently, a thorough description of the research methods (architectures selected, benchmarks and datasets) implemented is also be provided, so that the reader develops a correct understanding of the investigation.

3.1 Understanding Vision-Language Models - Revisited

The main purpose of this dissertation is to shed some light on the development of SoTA VL models. As mentioned in section 2.4, the trends within this subfield are in constant change, as scientists explore diverse approaches in the search for human level capabilities. This results in a wide variety of possible directions towards which steer VL research.

Even though pre-trained Transformers have proven to be the preferred base architectures in the field, there are numerous other design choices when it comes to developing a VL model. Specifically, this project focuses on these presented bellow.

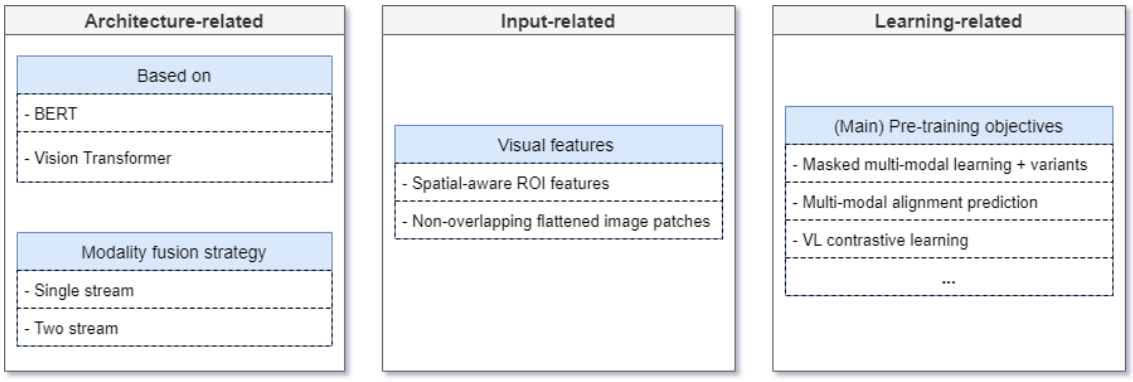

Figure 3.1: Subset of design choices studied in this dissertation [40].

An appropriate comparison of the available methods within the field would in turn accelerate its evolution, directing the scientists' resources to prioritize the most promising ones.

This project intends to carry out said comparison by selecting a current and representative subset of 5 SoTA VL architectures, to subsequently fine-tune and benchmark them on two common

tasks. The results of these benchmarks will then be used to derive how different combinations of design choices affect these models' performance.

3.2 Methods: Models selected

In order to ensure that this project provided a representative and current comparison of VL architectures, a thorough investigation on the available approaches published from 2019 onwards was conducted. Subsequently, the selected models were appropriately analysed, so that their comparison could be grounded on a solid understanding of their methods and design choices.

The following subsections summarize this process by covering each of the selected approaches to the necessary level of detail, highlighting the design choices and techniques applied by their authors during their development.

3.2.1 ViLBERT - 2019

The first model selected was ViLBERT, a two-steam BERT-based architecture published in 2019 [40].

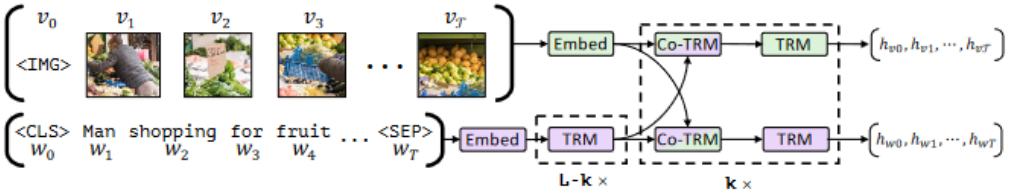

Figure 3.2: ViLBERT architecture as depicted in its original paper [40].

This architecture consists in two parallel BERT-style streams. The linguistic stream starts with a series of *transformer encoder blocks* (TRM) which extract intermediate representations from the inputs, while the visual features do not undergo any uni-modal processing. In order to subsequently perform the fusion, the authors introduced the idea of *co-attentional transformer modules* (Co-TRM), as depicted in figure 2.16. In them, given intermediate visual and linguistic representations $H_V^{(i)}$ $H_W^{(i)}$, the module computes query, key and value vectors, as presented in section 2.4.1. However, said keys and values are interchanged between modalities, allowing for the linguistic stream to pay attention to the visual features and vice-versa [40]. The output of said modules is fed into an additional pair of *transformer encoder blocks* as the final step in the modality fusion.

Model inputs. ViLBERT's authors opted for *bottom-up attention features* [6] as their visual inputs. These are extracted from the images by a *pre-trained object detector network* (Faster-RCNN), which localises N bounding boxes covering regions of interest, and produces an M dimensional descriptor for each of them. In order for the model to also consider the location of said objects in the picture, each descriptor is combined with a vector encoding the normalized coordinates of their respective box.

On the other hand, the linguistic stream operates on BERT-style inputs i.e. an aggregation of learned (sub-)word embeddings with both encodings of position and segment (representing which sentence the token belongs to, in the case that more than one is given e.g $\{question\}$ $\{answer\}$)

Special tokens, [IMG] and [CLS], are prepended to both the visual and textual inputs for them to be treated as the holistic (mean pooled) representation of each stream when processed. Other task-specific tokens such as [MASK] (for masking) and [SEP] (to serve as a partition cue among different input sentences), are also used.

Pre-training. ViLBERT was pre-trained using the *Conceptual Captions* [52] dataset, on two of the aforementioned tasks: *masked multi-modal learning* and *multi-modal alignment prediction*.

As part of the first objective, approximately 15% of the inputs are selected for masking. Masked words are replaced with the special [MASK] token 80% of the time, a random word 10%, and left unaltered the remaining 10%. In the case of masked visual features, they are zeroed out 90% of the time, and unaltered the rest. The model then is assigned to infer said inputs' labels, as described in section 2.4.

In the latter task, positive and negative text-image pairs are randomly selected from the dataset. The alignment of the visual and linguistic stream is determined by a final linear layer operating on the overall representation of the input. Said overall vector is computed via an element-wise product between the outputs h_{IMG} and h_{CLS} (mean pooled final representation of the two streams).

Architectural details. For the purpose of this investigation, the variant $ViLBERT_{BASE}$ was selected. Its visual and linguistic streams consist on 6 and 12 encoder blocks, with a hidden size

of 1024 and 768, as well as 8 and 12 attention heads each. This model is further comprised of 6 co-attentional blocks, with 8 attention heads and a hidden size of 1024.

Table 3.1: General overview of ViLBERT's design choices.

Architecture-related	
Based on	*BERT*
Modality fusion strategy	*Two-stream*
Input-related	
Visual features	*Spatial-aware ROI features (Faster-RCNN)*
Learning-related	
Pre-training objectives	*Masked MM learning / MM alignment prediction*
Pre-training datasets	*Conceptual Captions*

3.2.2 LXMERT - 2019

The second model selected was LXMERT, a direct successor of the ViLBERT architecture published only months later. [56].

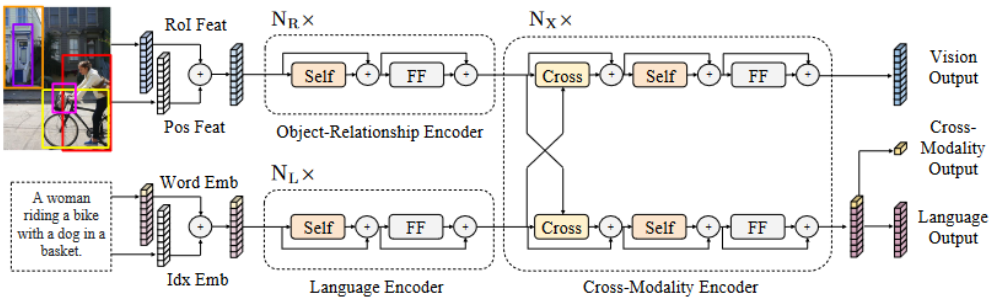

Figure 3.3: LXMERT architecture as depicted in its original paper [56].

As another BERT-style two-stream model, LXMERT follows a similar structure to the one presented above, while exhibiting key differences in its design: Uni-modal *object-relationship* and *language* transformer encoders are now included at the beginning of both streams, in contrast to the approach taken by ViLBERT's authors, who thought that this was not necessary for the extracted ROI visual features. On the other hand, the cross-modality encoder in LXMERT adopts a simpler structure compared to its predecessor's, with a *bi-directional cross-attention sub-layer*, for the information exchange among streams, followed by two *self-attention sub-layers*. The cross-modality encoder ends with two *feed-forward sub-layers* for further processing of the final outputs.

Residual connections are included in both the uni-modal and cross-modality encoders to ensure a more optimal gradient flow during back-propagation.

Model inputs. LXMERT uses the same type of *spatial-aware bottom-up attention features* as its predecessor. The only difference present, is that the projected vector which encodes the bounding-boxes' position is averaged with the projection of its respective descriptor, instead of solely summed.

On the other hand, the architecture's linguistic stream operates on a *simplified* version of BERT-style embeddings. LXMERT is not pre-trained on nor fine-tuned to any task which requires of multiple-sentence inputs, hence the authors decided to dispense with segment encodings.

The usage of task-specific and special tokens is also leveraged during LXMERT's training. In this case, a [CLS] token is prepended to each sentence, to be treated as the overall VL representation of both streams when processed. Other task-specific tokens such as [MASK] (for masking) and [EOS] (to serve as an end of sentence cue), are also used.

Pre-training. LXMERT was pre-trained using an aggregation of the following 4 large-scale VL datasets: *MS COCO* [37], *Visual Genome* [31], *VQA* [3] and *GQA* [26]. Said datasets allowed the application to 3 distinct tasks: *masked multi-modal learning*, *multi-modal alignment prediction* and *visual question answering*.

Following BERT's guidelines, approximately 15% of the inputs were selected for masking. However, in the case of masked image regions, LXMERT is now tasked with regressing its original input feature, in addition to predicting its detected object label (simmilarly to ViLBERT's pre-training).

As part of the second task, positive pairs' sentences are replaced with miss-matched ones with a probability of 0.5. The alignment of the visual and linguistic stream is determined by a final classifier operating on the overall representation of the input h_{CLS}. Whenever a positive sample which contains a question is drawn from the dataset, this overall representation is paired with a distinct classifier, which infers the desired answer.

Architectural details. In order to ensure a fair comparison, the variant $LXMERT_{DEFAULT}$ was selected due to their similar size. Its visual and linguistic uni-modal streams consist on 5 and 9 encoder blocks, with a hidden size of 768 and 12 attention heads. Said blocks feed into 5 cross-modality encoders, with self and cross-attention sub-layers of the same dimensions.

Table 3.2: General overview of LXMERT's design choices.

Architecture-related	
Based on	*BERT*
Modality fusion strategy	*Two-stream*
Input-related	
Visual features	*Spatial-aware ROI features (Faster-RCNN)*
Learning-related	
Pre-training objectives	*Masked MM learning / MM alignment prediction / VQA*[1]
Pre-training datasets	*MS COCO + Visual Genome + VQA + GQA*

[1] Only applied with positive-paired samples containing questions from VQA or GQA datasets.

3.2.3 Oscar - 2020

The third architecture selected was Oscar, the first single-stream model of this subset. [36].

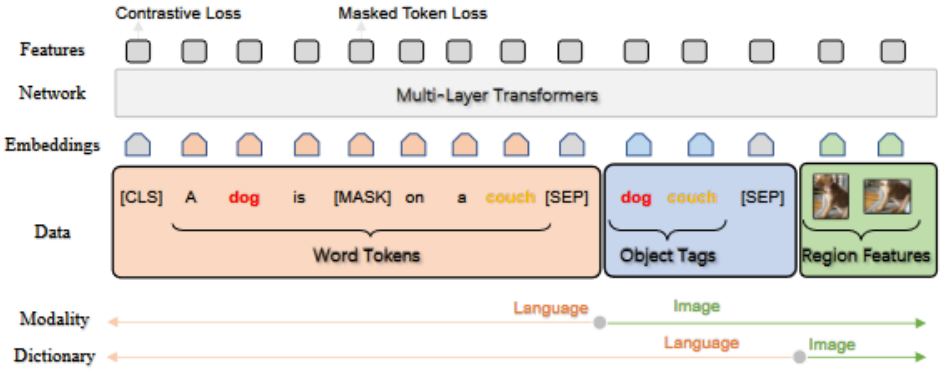

Figure 3.4: Oscar's architecture as depicted in its original paper [36].

As the first BERT-based single-stream model included in this investigation, Oscar's design breaks with the paradigm presented up to this point. Its authors dispense with any type of uni-modal processing as part of their approach, in favour of a single stack of standard transformer encoders comprised of a multi-headed attention layer, followed by a 2-layered MLP each, as illustrated in figure 2.13.

In order to further improve the alignment capabilities of the model, the authors introduced the idea of using high-confidence object tags as anchor points during pre-training. Said tags correspond to the categories detected in the input image by the feature extractor network (Faster-RCNN).

Model inputs. *Spatial-aware bottom-up attention features* were the de-facto choice for BERT-based VL architectures. However, Oscar developed its own method of encoding spatial information in the ROI descriptors: Instead of projecting the features and location embeddings to a common dimension for them to be added together, the authors decided to, for each descriptor, create a 6/4-dimensional vector $z = [x_L, y_L, x_R, y_R, (w, h)]$ containing the *coordinates* of the detected bounding box, and optionally its *width* and *height*. They would then be concatenated with their respective feature to form the final input.

The linguistic tokens are embedded using the same strategy presented in sub-section 3.2.1.

In terms of special inputs, a [CLS] token is added at the beginning of each sample, to be treated as the overall VL representation of both modalities when processed. Other task-specific tokens such as [MASK] (for masking) and [SEP] (to allow for multi-sentence constructions such as *[sentence(s) + object Tags + ROIs]*), are also used.

Pre-training. Oscar's authors collected data from the following large-scale VL datasets: *Conceptual Captions* [52], *MS COCO* [37], *Visual Genome-QA* [31], *VQA* [3], *SBU*, [47], *Flickr* [61], and *GQA* [26]. This allowed for the model's pre-training on only 2 distinct tasks: *masked language modelling with visual clues* and *multi-modal contrastive learning*.

The first objective presents clear differences with respect to the masking-based tasks previously introduced. In this case, only the *linguistic inputs* (words or object tags) are to be replaced using the [MASK] token, with a probability of 15%. The model is then tasked with inferring the masked inputs using the rest of the sentence/tags, as well as the information given by the sample's image.

As part of the second objective, given a positive input triple (sentence, tags, ROIs), the authors generate negative ones by randomly sampling unrelated tags, with a probability of 50%.

Subsequently, they apply a fully-connected (FC) layer on the output representation of the [CLS] token, to predict whether the triple given is polluted. The *contrastive loss* allows for the generation of similar VL representation of inputs containing non-polluted tags, while pushing apart the remaining ones.

Architectural details. $Oscar_{BASE}$ was the variant selected for this experiment. It follows BERT's design with a stack of 12 encoders, comprised of 12 attention heads and a hidden size of 768.

Table 3.3: General overview of Oscar's design choices.

Architecture-related	
Based on	*BERT*
Modality fusion strategy	*Single-stream*
Input-related	
Visual features	*Spatial-aware ROI features (Faster-RCNN)*
Learning-related	
Pre-training objectives	*MLM with visual clues / MM contrastive learning*
Pre-training datasets	*CC*** + *MS COCO** + *VG-GQA** + *SBU*** + *Flickr** + *VQA** + *GQA*$^{+}$

$^{+}$ Balanced train split.

* Train split.

** All splits.

3.2.4 ViLT - 2021

The fourth model included in this investigation was ViLT, which introduced several changes to the VLP design norm. [30].

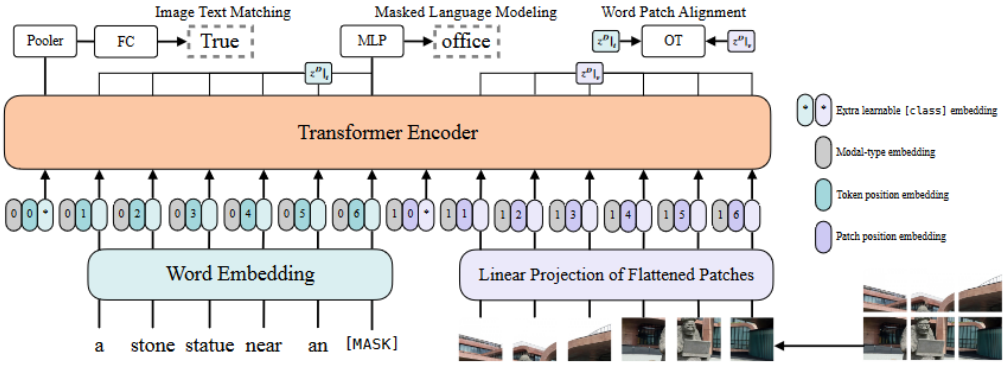

Figure 3.5: ViLT's architecture as depicted in its original paper [36].

ViLT supposed a paradigm shift in Vision-Language literature. At first glance, its structure can be compared to other single-stream approaches in the field; however, it presents two major differences:

This model bases its design in the Vision Transformer (ViT) [17], instead of the standard option to date, BERT. Said approaches exhibit the following architectural disparity: ViT-based encoders apply their normalisation prior to both the multi-headed attention and FF sublayers ("pre-norm") in contrast to the majority of transformers, which do the opposite ("post-norm").

On the other hand, ViLT's authors opted for a faster and parameter-efficient approach, completely disregarding the well-established *ROI features* in favour of *non-overlapping flattened image patches* as their visual inputs. This allowed for a **x10** speed-up in comparison to the previously introduced SoTA models.

Model inputs. ViLT operates on *non-overlapping flattened image patches*, as described in the paragraph above. This new approach to encoding visual inputs works as follows: The image $I \in \mathbb{R}^{C \times H \times W}$, where C, H and W are the channel, height and width dimensions, is sliced into square-shaped patches and flattened to $v \in \mathbb{R}^{N \times (P^2 \cdot C)}$, with N and P^2 being the number of individual slices $\frac{H \cdot W}{P^2}$, and the area (in pixels) of each patch. This representation is linearly projected to match the dimensions of both the *position* (which encodes the absolute location of each patch within the input sequence) and *modal-type embeddings* (specifying their belonging to the visual modality), which are then added to form the final input.

The linguistic tokens are encoded using the standard *word* and *position* embedding matrices, to subsequently be summed with their respective *modal-type embeddings*.

A [CLS] token is added at the beginning of both modalities. However, the one prepended to the visual modality merely works as a separator, while the remaining one is interpreted as the VL representation of the whole input after processed. Other task-specific tokens such as [MASK] (for masking) are also used.

Pre-training. ViLT's authors created their own corpus by collecting data from the 4 datasets: *Conceptual Captions* [52], *MS COCO* [37], *Visual Genome* [31], *SBU*, [47]. This allowed for the

model's pre-training on only 2 commonly applied tasks: *masked language modelling* and *multi-modal alignment prediction*.

In contrast to the standard practice in the field, ViLT's masking-based pre-training task does not rely on the visual inputs at all. Instead, the model uses a final MLP operating over the textual subset of the final representations to perform its predictions. This may seem counterproductive since the objective is not enforcing any kind of cross-modality interaction; However, since the model's weights are initialised from a pre-trained ViT, they lack the *linguistic understanding* that BERT-initialised models have (ViLBERT, LXMERT, Oscar...) hence, this task allows ViLT to learn it during pre-training.

On the other hand, the second objective stimulates said interactions at both the modality and token level: The authors randomly replace the image of positive input pairs with 50% probability, and use a final FC layer operating over the overall VL representation to predict their alignment. Additionally, a *word-patch* alignment term is added to this task's loss function. This is calculated via the inexact proximal point method for optimal transports (IPOT) [60] over the textual and visual subsets.

Architectural details. $ViLT_{BASE}$ was the variant benchmarked as part of this investigation. It is the direct competitor of Oscar due to its size, with 12 ViT-style encoders, comprised of 12 attention heads and a hidden size of 768 each.

Table 3.4: General overview of ViLT's design choices.

Architecture-related	
Based on	*Vision Transformer (ViT)*
Modality fusion strategy	*Single-stream*
Input-related	
Visual features	*Non-overlapping flattened image patches*
Learning-related	
Pre-training objectives	*Masked language modelling / MM alignment prediction*
Pre-training datasets	*Conceptual Captions + MS COCO + SBU + Visual Genome*

3.2.5 ALBEF - 2021

The fifth and last model evaluated as part of this dissertation was ALBEF, which not only implements an intricate combination of the previously described design choices, but also introduces several new ones that offer great improvements to its performance [35].

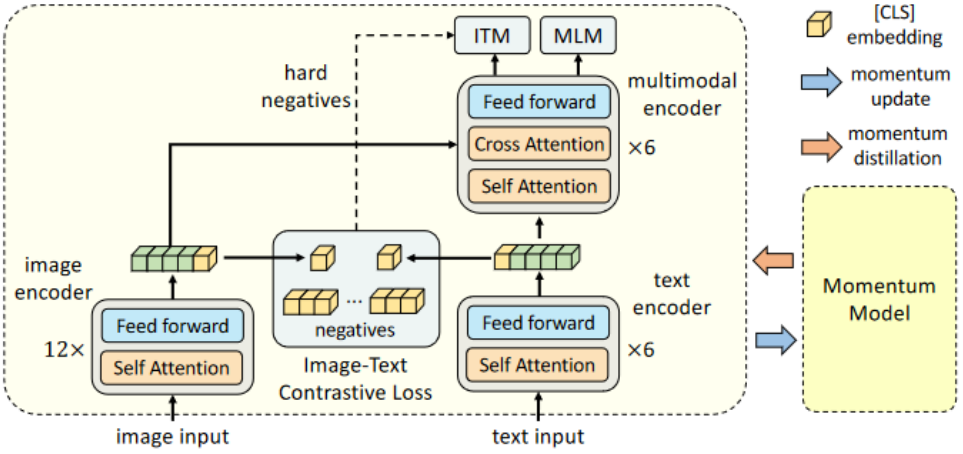

Figure 3.6: ALBEF's architecture as depicted in its original paper [35].

ALBEF is the first approach to leverage the advantages of the two-stream paradigm by implementing a *hybrid* BERT/ViT-based architecture. Its linguistic and multi-modal modules are standard transformer encoders, initialised from BERT's pre-trained weights, while its visual modules are based on and initialised from a pre-trained ViT. This strategy mitigates the problem of weights initialisation, where BERT-based architectures lacked visual understanding and ViT-based ones needed of extensive language pre-training to be competent.

The modality fusion strategy implemented in ALBEF drifts away from the present trends towards a classical transformer decoder-like construction. In this case, the multi-modal's self attention sub-layer operates on the intermediate representation of the textual stream. Subsequently, the interactions between the two streams are performed in the cross-attention sub-layer, where the *query* vectors are extracted from the textual representations, while the *keys* and *values* come from the visual encoder's output. This allows for the linguistic stream to pay attention to the image patches, but not in the opposite way.

The model's authors additionally introduced two well-know strategies to its pre-training: the usage of *contrastive learning* to align the uni-modal representations prior to their fusion, and *knowledge distillation* from an ensemble to further improve the performance.

Model inputs. ALBEF operates on ViT-style visual inputs, as described in sub-section 3.2.4.

On the other hand, the linguistic tokens are encoded using standard BERT-style embeddings, which were also previously explained.

A [CLS] token is added at the beginning of both modalities, and treated as the overall representation of each individual stream after being processed by their respective uni-modal encoder. The one prepended to the input sentence is further refined by the multi-modal encoder, finally transforming it into the overall VL representation of the whole input. Other task-specific tokens such as [MASK] (for masking) are also used.

Pre-training. ALBEF is pre-trained using 5 VL datasets: *Conceptual Captions* [52], *MS COCO* [37], *Visual Genome* [31], *SBU* [47] and *Conceptual 12M* [11]. Said datasets allowed for the application of three distinct tasks: *image-text uni-modal contrastive learning*, *masked language modelling with visual clues* and *multi-modal alignment prediction*.

The first mentioned task was one of ALBEF's novel contributions to the field: for every positive/negative text-image pair, the model is tasked with learning a similarity function that dictates how aligned their uni-modal representations are. The contrastive loss ensures that positive pairs will have resembling uni-modal representations and vice-versa, easing the fusion process.

ALBEF's second objective has an identical setup to what was described in section 3.2.3, as only the linguistic tokens are selected for replacement/masking.

Finally, the *multi-modal alignment prediction* task also follows the standard practice implemented by previous models. Given positive/negative text-image pairs, the model learns a FC layer over the final representation of the linguistic [CLS] token. This layer, followed by a softmax, performs the final binary classification.

These tasks were implemented using large-scale web-based datasets, which tend to include a significant amount of noise in their ground truth i.e. **for a given caption, there may be another acceptable (or even better) combination of words that also describe its corresponding image**. On the other hand, the random nature of some tasks may also negatively influence the learning process i.e. **the text randomly replaced to form a "negative" pair may correctly entail the input image**. This is why ALBEF makes use of the knowledge distillation strategy, where the model does not learn from the ground truth directly, but from the predictions of a "teacher". In this case, the teacher is an continuously-evolving ensemble of exponential-moving-average versions of the uni-modal and multi-modal encoders.

Architectural details. $ALBEF_{BASE}$ was the variant benchmarked as part of this investigation. Its linguistic and multi-modal encoders contain 6 BERT-style encoders , comprised of 12 attention heads and a hidden size of 768 each. The visual stream is a stack of 12 ViT-based encoders of the same dimensions.

Table 3.5: General overview of ALBEF's design choices.

Architecture-related	
Based on	*Vision Transformer (ViT) + BERT*
Modality fusion strategy	*Two-stream*
Input-related	
Visual features	*Non-overlapping flattened image patches*
Learning-related	
Pre-training objectives	*MLM + visual clues / MM contrastive learn. / MM align. prediction*
Pre-training datasets	*Conceptual Captions + MS COCO + SBU + VG + Conceptual 12M*

3.3 Methods: Dataset and Downstream tasks

The coming subsections will present the benchmarking tools utilized to carry out this investigation. More precisely, the dataset used, as well as the selected downstream tasks will be discussed.

3.3.1 Dataset: Places365

Places365, in its standard version, is a subset of the MIT Places dataset: a 10M image database for deep scene recognition [62]. The original repository contains 10 million scene photographs, labeled with 434 semantic categories which, according to its authors, comprises about 98% of the type of places that a human can encounter in the world.

The main motivation behind the usage of this specific dataset was its inter and intra-category diversity: as a web-based database, its images were gathered via internet queries of around 900 pre-defined classes. However, in order to also ensure an optimal coverage among different views of one same class/place, the authors coupled said queries with 696 common English adjectives, as depicted in the figure bellow.

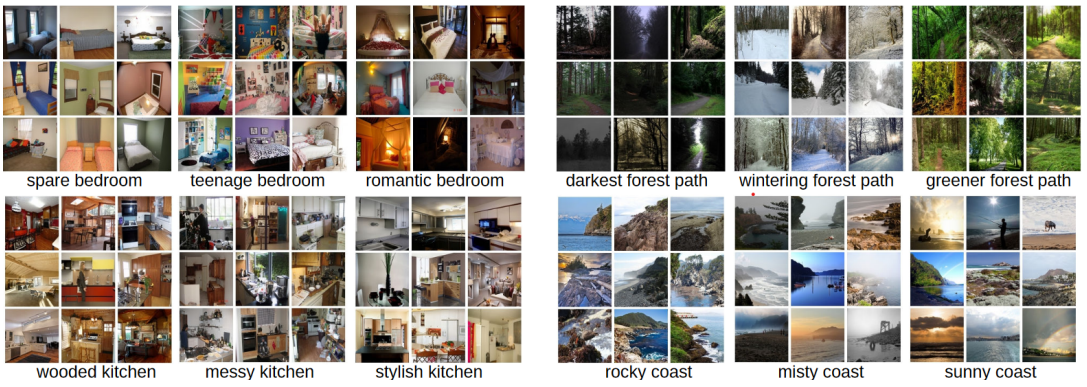

Figure 3.7: Examples of queries corresponding to 4 distinct categories: ***kitchen, bedroom, forest path and coast*** [62].

As stated at the beginning of this section, the subset of this image collection used as benchmark was the so-called **places365-Standard-small**. It contains a selection of 2.1M, 256×256 images, corresponding to 365 categories out of the original 434. A higher-resolution version is available in its original website; however, the lower-resolution variant proved to be more suitable due to the constrained computational resources and time that this project was completed under.

Table 3.6: Overview of Places365-Standard and its original splits.

	Train	**Validation**	**Test**
# Images	$1,803,460$	$36,500$	$328,500$
# Images per category	$3,068$ - $5,000$	100	900

Nonetheless, it is critical to note that **the ground truth annotations for the test set were not released to the public**. Instead, its authors created an evaluation server, to which you could submit your models' top-5 classification predictions. This presented a clear problem, as it reduced the possibilities of downstream tasks to only image classification.

On top of that, both the dataset and its evaluation server became unavailable during the full time phase of this project (from mid-July onwards), due to maintenance reasons. Every attempt

of contacting the authors was unsuccessful, and with only one month and a half left, I decided not to switch to another database as this would mean re-factoring a considerable part of the code to adapt to it.

To overcome these two major problems, I created task-specific test splits by extracting balanced subsets of training samples while avoiding any form of data leakage. The details of said splits will be introduced in the following subsections

3.3.2 Downstream tasks

The correct selection of benchmarks was also a crucial part of this project. For that reason, a preliminary investigation into the downstream tasks applied in the selected models' papers was conducted. This would allow for an identification of tasks that are thought as representative of VL intelligence according to scientists in the field. The results of said investigation are summarized in the table bellow

Table 3.7: Downstream tasks present in the selected models' papers.

	VQA	**VE**	**VCR**	**RE**	**ITR**	**IC**	**NOC**
ViLBERT	✓	✗	✓	✓	✓	✗	✗
LXMERT	✓	✗	✓	✗	✗	✗	✗
Oscar	✓	✗	✓	✓	✓	✓	✓
ViLT	✓	✗	✓	✗	✓	✗	✗
ALBEF	✓	✓	✓	✓	✓	✗	✗

VQA: Visual question answering

VE: Visual entailment

VCR: Visual commonsense reasoning

ITR: Image-text retrieval

IC: Image captioning

NOC: Novel object captioning

On top of this, another important factor to consider is the compatibility of each task with the selected dataset. The majority of objectives included in the table above require the usage of specific sample constructions, only available in their respective databases i.e. the visual question answering objective requires of *[image + question + answer]* type samples, not available in the Places365 collection.

According to these constraints, the downstream tasks designated for the purpose of this investigation were ***Image Classification*** and ***Text to Image Retrieval***.

3.3.2.1 Downstream task: Image Classification

As stated in section 2.4, one of the factors that motivate the visual and linguistic modalities fusion is the hypothesis of their ability to generate representation spaces which are more disentangled.

With the addition of a purely-visual task to the benchmarking of these pre-trained VL models, we could not only compare the quality of their extracted representations against each other, but also against architectures that do not include any language modeling in their designs, such as CNNs. This would grant scientists further information to debunk or support the previously mentioned hypothesis.

In order to successfully apply said objective, a balanced test set with the same characteristics as the original was created, by extracting 900 samples of each category from the train split. This custom variant of the dataset is summarized in the table bellow.

Table 3.8: Overview of the custom splits created as part of the image classification task.

	Train	**Validation**	**Test**
# Images	$1,474,960$	$36,500$	$328,500$
# Images per category	$2,168$ - $4,100$	100	900

3.3.2.2 Downstream task: Text to Image Retrieval

Text to image retrieval is one of the most popular tasks in the VL sub-field. This is because it requires of models to not only exhibit *deep linguistic understanding* of complex queries but also *advanced visual abstraction* capabilities. Moreover, the general nature of this objective allows it to be implemented using a wide variety of datasets, shifting the importance between the previously mentioned aptitudes in a flexible manner i.e. databases with intricate queries (e.g. dense image captions) require of superior language processing capabilities, while a deeper visual understanding is vital for image collections with high inter and intra-class diversity.

Due to the aforementioned characteristics, the **Category-based Image Retrieval** benchmark forms a perfect combination with our selected dataset: On the one hand, said deep linguistic understanding is not enforced by long and complex queries, but by the existence of lexically related classes that ultimately represent different locations e.g *hokey arena, performance arena and rodeo*

arena. On the other hand, Places365 contains numerous distinct views belonging to each scene type, which require of advanced visual abstraction capabilities for their successful categorization.

Following the implementation trends present in papers that study retrieval, I created a smaller evaluation set by randomly selecting 10 samples belonging to each category from the previously presented custom test split.

Table 3.9: Overview of the custom splits created as part of the retrieval task.

	Train	**Validation**	**Test (3.6k)**
# Images	$1,474,960$	$36,500$	$3,650$
# Images per category	$2,168$ - $4,100$	100	10

4 TECHNICAL CHAPTER 2: EXPERIMENTAL SET-UP

After introducing the methodology necessary to carry out the work present in this dissertation, this next subsection summarizes the experimental set up of this investigation. A description of the selected architectures' adaptation strategy to both benchmarks is provided. Furthermore, details regarding tasks' implementation (both training and inference), as well as hyperparameter choice are discussed in order to allow the reproduction of results if desired.

4.1 Image Classification: Experimental Set-up

Adaptation method. In order to ensure a fair comparison among the architectures, they were adapted to the image classification task following a *quasi-identical* strategy. The architectural adaptation of the selected models was as follows:

- **ViLBERT:** Two fully-connected layers, with GeLU activation [23] and layer normalization [7] (post-activation) were learned on top of the *mean-pooled representation of the **visual stream***.
- **LXMERT:** Two fully-connected layers, with GeLU activation and layer normalization (post-activation) were learned on top of the *mean-pooled representation of the **entire input***.
- **Oscar:** Two fully-connected layers, with GeLU activation and layer normalization (post-activation) were learned on top of the *mean-pooled representation of the **entire input***.
- **ViLT:** Two fully-connected layers, with GeLU activation and layer normalization (pre-activation) were learned on top of the *mean-pooled representation of the **entire input***.
- **ALBEF:** Two fully-connected layers, with GeLU activation and layer normalization (post-activation) were learned on top of the *mean-pooled representation of the **entire input***.

The dimensionality of the attached classifiers is **proportional** among each other i.e. *twice* the hidden size of the token they operate over. On the other hand, the layer normalization's position on each model was influenced by their authors' guidelines, who already implemented similar constructions in their respective paper.

Model inputs. The visual inputs utilized by **ViLBERT**, **LXMERT** and **Oscar** are *Bottom-up-attention mean-pooled features*, of size $N = 2048$ extracted from 36 objects detected in each image. This process was carried out by a *Faster-RCNN* network, as described in Anderson et al. [6]. **ViLT** and **ALBEF** did not require of any visual pre-processing, as the slicing/flattening was performed by the architecture itself.

On the other hand, no linguistic inputs were necessary since this is a purely visual task, hence sequences only containing special tokens such as [CLS] and [SEP] (when the architecture required of it) were used.

It is worth noting that **Oscar's** authors make use of *object tags* (as introduced in section 3.2.3) during the fine-tuning process, to further align their visuo-linguistic representations. However, since said representations were fixed during this experiment (more information in the paragraph bellow), and due to the uni-modal nature of the task, these tags would only grant the architecture an effective advantage over its competitors. Therefore, in compliance with the previously mentioned reasons, **the use of object tags was discarded**.

Training details. To adequately analyze the selected models' capabilities, their *pre-trained weights were frozen* while only the classification heads were updated during the fine-tuning process. In consequence, their performance discrepancies when tested would solely be influenced by their ability to extract meaningful representations, given by their respective pre-training strategies and design choices, without further adaptation to the task/dataset in hand.

All the architectures were trained for 50 epochs and a batch size of 32 samples, using PyTorch's default configuration of the *AdamW* optimizer [39] and *Cross-entropy* loss as the objective to minimize. Said loss function is defined in the equation below, where t_i is the ground-truth label and p_i is the Softmax probability for the i_{th} class.

$$Loss_{CCE} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for } n \text{ classes}, \tag{4.1}$$

After performing a *random hyperparameter search*, **ViLBERT, LXMERT, Oscar** and **ALBEF** were found to best perform when fine-tuned using a *maximum learning rate* of $\eta = 3 \cdot 10^{-4}$, while **ViLT** benefited from larger updates with $\eta = 3 \cdot 10^{-3}$. Said values were updated throughout the fine-tuning stage by two scheduling techniques:

- **Linear warm-up**, which consists on starting the training process with a learning rate *lower* than ultimately intended, to then iteratively *increase* it until reaching the desired maximum. In this case, it was implemented over 10% of the training steps as a complement to AdamW. This optimization algorithm supports its updates with statistics calculated over the training batches, which may be noisy at the beginning due to its inability to capture the real underlying distribution of the dataset. By using a smaller η during the early training stages we mitigate the negative effect of said noisy updates.
- **Linear decay**, which allows for better exploitation capabilities when traversing the objective function's landscape in the final epochs. It does so by *linearly* decreasing η as the training progresses, ensuring that the optimization algorithm will not *"overshoot"* the minima via excessively large updates.

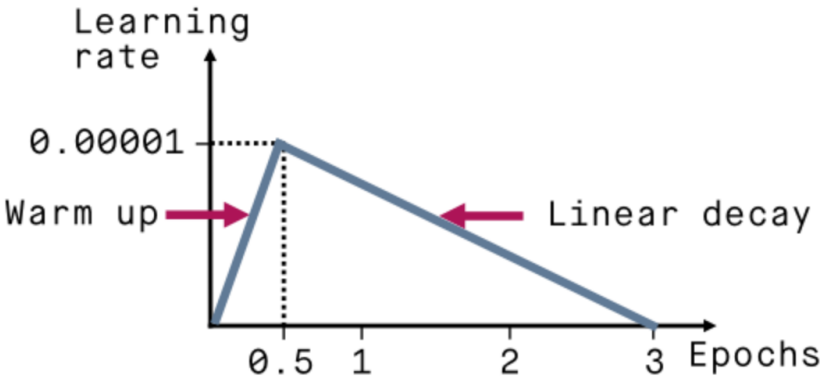

Figure 4.1: Depiction of the learning rate scheduling techniques utilized during the fine-tunning process (the values present in the figure are for explanation purposes only) [1].

Inference details. The models were assessed on the Places365 test split presented in table 3.8, reporting *Top 1 accuracy* as the designated metric. In order to allow for an unbiased comparison with other SoTA models, performance on the validation set will also be disclosed.

Table 4.1: Overview of the first experiment (Image classification).

		Training tools		Hyperparameters		
	LayerNorm pos.	Optimizer	Loss func.	η	# Epochs	Batch size
ViLBERT	post-act.	AdamW	CCE	$3 \cdot 10^{-4}$	50	32
LXMERT	post-act.	AdamW	CCE	$3 \cdot 10^{-4}$	50	32
Oscar	post-act	AdamW	CCE	$3 \cdot 10^{-4}$	50	32
ViLT	pre-act.	AdamW	CCE	$3 \cdot 10^{-3}$	50	32
ALBEF	post-act.	AdamW	CCE	$3 \cdot 10^{-4}$	50	32

CCE: Categorical Cross-entropy.

4.2 Text to Image Retrieval: Experimental Set-up

Adaptation method. Due to the high similarities in approach that the two objectives share (see *Training details* for more information), the selected models' architecture was adapted in an identical manner to the aforementioned in section 4.1. Nevertheless, **ViLBERT's** adaptation strategy did require of an adjustment for its application to this task: the model's final retrieval head now operates on the *mean-pooled representation of the* ***entire input*** to account for the relationship between streams.

Model inputs. The images utilized as part of this experiment share the same pre-processing as described for the previous objective.

On the other hand, the linguistic inputs were embedded following the instructions given by the models' authors, as presented in section 3.2. No object tags were utilized during the fine-tuning of Oscar since their only function should be assisting in the modalities alignment process, which is not present at this stage.

Training details. Adhering to a similar strategy as the one present in other VL publications, retrieval is formulated as a ***binary classification*** *problem* during training: given a positive text-image pair (category + image), the script randomly selects a different class to form an unaligned one, with 50% probability. The architectures' final MLP is tasked with learning a similarity function between the modalities, which will predict whether a given pair is aligned or not.

The architectures' pre-trained weights were not modified during the fine-tuning process, in accordance with the reasoning presented in the sub-section above. On the other hand, their respective retrieval heads were trained for 50 epochs and a batch size of 32 to minimize the *Binary Cross-entropy* function using *AdamW* as their optimizer. Said objective function is depicted in the equation bellow, where t_i is the ground-truth label and p_i is the Softmax probability for the i_{th} sample.

$$Loss_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(t_i \log(p_i) + (1 - t_i)\log(1 - p_i)), \quad \text{for } n \text{ samples,} \tag{4.2}$$

The *learning rate* maximum values and *schedulers* were as described for the previous task.

Inference details. The assessment process was carried out as follows: using the test split described in table 3.9, for each query (class) in the dataset, the models iterated through the entire collection of images, calculating their alignment probability, which would subsequently be used for ranking. The average *Top K precision* and *recall*, with $K = [1, 5, 10]$, over all classes were designated as the performance metrics of this task.

Table 4.2: Overview of the second experiment (Image to text retrieval).

		Training tools		Hyperparameters		
	Layernorm pos.	Optimizer	Loss func.	η	# Epochs	Batch size
ViLBERT	post-act.	AdamW	BCE	$3 \cdot 10^{-4}$	50	32
LXMERT	post-act.	AdamW	BCE	$3 \cdot 10^{-4}$	50	32
Oscar	post-act	AdamW	BCE	$3 \cdot 10^{-4}$	50	32
ViLT	pre-act.	AdamW	BCE	$3 \cdot 10^{-3}$	50	32
ALBEF	post-act.	AdamW	BCE	$3 \cdot 10^{-4}$	50	32

BCE: Binary Cross-entropy.

5 TECHNICAL CHAPTER 3: EXPERIMENTS RESULTS AND DISCUSSION

The following chapter presents the results obtained as part of the experiments previously described. Subsequently, the selected architectures are re-visited in ascending order of performance, in order to critically analyse the components and design choices which mostly influenced their outcomes.

5.1 Image Classification: Results and Discussion

The results obtained during the assessment process described in section 3.3.2.1 are summarized in the table bellow.

Table 5.1: Summary of the first experiment's results (Image classification). The highest performing visuo-linguistic approach appears in **bold**.

		Top 1 accuracy	
		Places365 Validation Set	Places365 Test Set*
Visuo-linguistic methods	ViLBERT	43.1%	43.49%
	LXMERT	43.26%	44.72%
	Oscar	39.41%	40.3%
	ViLT	46.65%	47.19%
	ALBEF	**52.01%**	**52.3%**

* Custom test set described in 3.8.

The Image Classification task heavily relies on the networks' ability toz recognise, not only deeply *understand* visual content, but also successfully *align* it with its semantic meaning, in order to generate representations which are easily-distinguishable between classes. This discussion will focus on identifying the approach-specific combination of mechanisms which allow for said aptitudes:

Oscar. As the lowest-performing approach, Oscar represents a poor combination of design choices for a problem where extensive visual understanding is essential.

In terms of its *input-related* design decisions, the choice of a heavier image pre-processing strategy was expected give the architecture an advantage over ViT-based approaches, but proves to be ineffective on its own.

As a model *based on and initialised from BERT*, it possesses an advanced understanding of language but is deficient in optical processing power. However, this drawback inherent to linguistically-initialised models does not limit the performance (to this extent) of other approaches such as LXMERT.

The reason behind this behaviour is thought to be the *lack of objectives* focused on stimulating said standalone visual understanding capabilities during Oscar's pre-training: on the one hand, *masked language modelling with visual clues* does not force the architecture to infer the replaced tokens by using information from the input image, specially since, as a linguistically-initialised architecture, its natural choice is to search for context in the rest of the *input sentence* and/or *object tags*. On the other hand, *multi-modal contrastive learning* has proven to be an effective tool in other approaches, yet its primary purpose is to build inter-modality relationships, which does not grant any substantial benefits by itself for this specific task.

ViLBERT. The performance improvement shown by this architecture with respect to Oscar's reaffirmed the logic presented in the paragraphs above.

ViLBERT shares the same type of visual pre-processing as the previous approach, while surpassing it by 4 and 3% in classification accuracy during validation and inference respectively. This supports the idea that *bottom-up attention features* are not a limiting factor for VL efficacy, whilst structural and learning-related choices are.

In terms of architectural discrepancies, the success of ViLBERT over Oscar could be attributed to its two-stream modality fusion mechanism, yet there is evidence according to the results presented which refutes said conjecture: To begin with, the ability to generate effective language-aligned visual representations, while not entirely, is highly influenced by the models pre-training strategy, hence Oscar would be expected to have an advantage as a consequence of leveraging contrastive learning. Furthermore, higher visual processing capabilities are of greater importance in this specific task, which does not directly require of an effective modality fusion.

This model's superior results come as a consequence of mitigating the poor visual understanding capabilities that its BERT-initialised weights posses, via specific pre-training. The authors decision of randomly masking ROI features during the learning process forced the architecture to learn how to better exploit said modality.

LXMERT. This two-stream approach only presents slight performance improvements with respect to its predecessor, which due to their extensive similarities can mainly be attributed to two reasons:

LXMERT's modality fusion module is simpler in terms of number on parameters with respect to the previously described. Furthermore, ViLBERT's entire visual stream contains 12 individual transformer encoders (including co-attentional layers) in contrast to LXMERT, which is only comprised of 10. In consequence, ViLBERT's inputs go through a more complex processing overall, which according to the results, does not grant the architecture any performance advantages. This indicates that LXMERT's object-relational uni-modal encoders offer a boost in visual encoding abilities.

On the other hand, the decision of not only predicting the label of the zero'ed out ROIs during *Masked Multi-Modal Learning* but also regressing the input feature clearly grants LXMERT an advantage in its understanding of optical inputs.

ViLT. As the first ViT-based architecture, ViLT is able to surpass every BERT-based model in this task, without any explicit visual-centered pre-training and a simpler image pre-processing strategy.

This is mainly due to its inherent ability to process visual data, given by its weights initialisation. ViLT was initialised with weights from the vision transformer ViT-B/32 pre-trained on ImageNet, which gives the model an advantage over BERT-initialised architectures.

The rest of ViLT development was focused on granting the architecture linguistic/cross-modal capabilities, which do not directly influence its performance in this task. This suggests that ViT weight initialisation is more effective than visual-centered pre-training objectives (masked multi-modal learning with feature regression) for the development of optical processing capabilities.

ALBEF. This two-stream model model obtained the highest performance out of the benchmarked architectures due to the following reasons:

To begin with, its visual encoder is based on and initialised from a vision transformer, which grants inherent visual processing capabilities to the model, as explained above. However, ALBEF's authors decided to make use of ViT-B/16, which operates over images patches of size 16×16, in contrast to ViLT, which utilized 32×32 slices. This reduction in size and increase of granularity in terms of number of patches and computational complexity, is one of the factors which allow for such high performance.

On the other hand, ALBEF's authors exploit the fact that ViT-based architectures operate directly over images instead of extracted features (which tend to be pre-computed for efficiency), by applying augmentations during pre-training. This had already proven to be highly effective in boosting generalization and performance in purely visual models.

In light of the above, the key hypotheses developed according to the results obtained are the following:

- **Input feature-related (Visual Features):** VL models which make use of ViT's simpler visual pre-processing strategy outperform architectures operating over the more computationally expensive Faster-RCNN features in tasks which require of heavy visual understanding (such as image classification).
- **Architecture related (Base Models):** ViT-based architectures achieved much better results (even without any explicit visual-centered VL pre-training) in the image classification task because they were initialised with weights trained on large-scale image datasets. On the other hand, BERT-based architectures lack visual understanding in comparison since their weights were initialised from pure linguistic pre-training.
- **Learning-related (VL Pre-training Objectives):** BERT-based architectures highly benefit from pre-training tasks that stimulate their visual understanding, such as *masked multimodal learning*. Moreover, regressing the masked inputs' features provides a performance boost with respect to only predicting its label. Other pre-training tasks which "make use" of the visual inputs indirectly such as *masked language modelling with visual clues* do not offer the same benefits. This is thought to be due to BERT's linguistic nature, which causes

it to be more prone to look for context in the rest of the input sentence (or object tags in the case of Oscar) instead of utilizing the information of the visual stream.

As stated in section 3.3.2.1, the selection of *image classification* as one of the benchmarks in this project allows for the comparison with architectures that do not include any linguistic data in their training. This would help in the process of confirming or debunking the hypothesis which states that language-aligned visual representations offer a boost in performance in purely visual tasks.

For the purpose of this comparison, the results obtained by 4 different CNN architectures were extracted from the Places2 original paper [62], and can be seen in the table bellow.

Table 5.2: Summary of the results obtained by the CNNs presented as baseline in the places2 paper [62]. The highest performing purely visual method is highlighted in red.

		Top 1 accuracy	
		Places365 Validation Set	Places365 Test Set*
Non-linguistic methods+	Places365-AlexNet	53.17%	-
	Places365-GoogLeNet	53.63%	-
	Places365-VGG	**55.25%**	-
	Places365-ResNet	54.74%	-

+ The presented non-linguistic methods and their results were extracted from the Places2 original paper [62].

* Custom test set described in 3.8.

According to these results, the *non-linguistic* methods outperform every VL architecture benchmarked as part of this investigation. However, it is worth noting that they have three major discrepancies in their training set up. To begin with, the presented CNNs were trained on the **full Places365 training set**, which contains $\mathbf{328,500}$ images more than the reduced set used in this dissertation. Moreover, due to time constraints, the selected architectures were only fine-tuned for $\mathbf{50}$ epochs, while the places365-CNNs were trained for $\mathbf{90}$. Finally, the weights of the VL models were not optimized during the fine-tunning process but only their classification heads, which is a considerable disadvantage with respect to the presented CNNs, that were trained entirely.

Taking into account these three aspects, in conjunction with the low margin among the best-performing VL architecture and the non linguistic methods ($\mathbf{1.16\%}$, $\mathbf{3.24\%}$ and $\mathbf{2.18\%}$ with respect to AlexNet, VGG and in average respectively), it is strongly suggested that linguistically

aligned features do assist in purely visual tasks. In order to prove this, the VL architectures could be trained sharing the same setup as presented in the paragraph above, yet this is out of the scope of this project.

5.2 Text to Image Retrieval: Results and Discussion

The results obtained as part of the benchmarking process described in section 3.3.2.2 are summarized in the table bellow.

Table 5.3: Summary of the second experiment conducted (Text to Image Retrieval). The highest performing visuo-linguistic approach appears in **bold**.

	Top-K Precision			Top-K Recall		
	P@1	P@5	P@10	R@1	R@5	R@10
ViLBERT	30.68%	23.45%	19.8%	3.06%	11.72%	19.8%
LXMERT	46.57%	40.32%	33.72%	4.6%	20.16%	33.72%
Oscar	1.6%	1.64%	1.34%	0.16%	0.82%	1.34%
ViLT	**50.1%**	**44.7%**	**37.83%**	**5.01%**	**22.35%**	**37.83%**
ALBEF	47.1%	40.65%	34.27%	4.71%	20.32%	34.27%

As a counterpart to the *Image classification task*, which mostly relied on the image representations, in the *Text to Image Retrieval* objective the models are also required to be capable of successfully understanding the relationship between the linguistic and visual inputs. However, It is worth noting that, this ability of recognising inter-modal relationships is highly influenced by how good the uni-modal representations are in the first place i.e. the more information a model can extract from both images an sentences, the easier it will be for it to infer the underlying relationship among them. This is why, even though the following discussion will focus on how the different design choices influence each architecture's *cross-modal capabilities*, the factors presented in section 5.1 also played a crucial role in the achievement of these results.

Oscar. This model's performance is well bellow the average achieved among its competitors, and while it is true that the dataset's complex visual nature makes it difficult for a BERT-based architecture to perform well (due to its weak image processing capabilities), it is not the main reason behind this behaviour.

As stated in section 4.2, retrieval was formulated as a binary classification problem during training, hence *accuracy* was used to keep track of the model's improvement. Oscar was not able to achieve more than $\mathbf{56\%}$ accuracy throughout the fine-tuning process while every other model was reaching to at least $\mathbf{90\%}$. This motivated the hypothesis that Oscar was under-performing due to its inability to understand the input sequence when *no object tags* were included.

In order to prove said hypothesis Oscar was re-trained, now including the $\mathbf{10}$ object tags with the highest confidence score detected in each image. However, the same outcome was reached, refuting the conjecture.

After further investigation, another hypothesis was proposed: The ability of VL models to understand the relationship between the modalities particularly depends on a correct selection of pre-training strategies e.g. *MM alignment prediction*. Oscar was supposed to learn this mainly from the *"MM contrastive learning"* task i.e. "negative triples = no relationship, positive triples = strong relationship". However, since Oscar's negative triples were generated by sampling random *object tags*, the model had only learned to "look" for this relationship among them and the rest of the input, not among the two modalities i.e. sentence and image.

To further test this, the model was re-trained, sampling random polluted object tags for negative sentence-image pairs. This time, after $\mathbf{50}$ epochs, Oscar achieved $\mathbf{98\%}$ accuracy, yet since it did not have the knowledge to discern between positive and negative class-image pairs, it could not be tested like the others. I believe that Oscar could learn to understand this relationship if trained entirely (without freezing its weights up to the retrieval head) but this is out of the scope of this project.

ViLBERT. It is the worst-performing two-stream model, regardless of having the most complex cross-modal encoder in terms of both layers and parameters.

However, when analysing its design choices, it can be seen that the majority of them are shared with better-performing architectures. This indicates that ViLBERT's low results are caused by two main reasons: As stated at the beginning of the section, the models' ability to extract useful information from each individual modality boosts their cross-modal understanding, hence ViLBERT's low visual processing capabilities (caused by the design choices presented in the section above) also affect its performance in this task. On the other hand, this model was pre-trained on the small-

est corpus out of all the architectures selected. This enormous difference in pre-training data was the most detrimental factor of this model's performance.

LXMERT. ViLBERT's direct successor clearly outperforms it in terms of cross-modal capabilities, achieving a much larger margin among their results with respect to the previously presented.

This difference in the margin between their results (less than $\mathbf{1.3}\%$ in test accuracy compared to $\mathbf{16.10}\%$ and $\mathbf{1.54}\%$ in top 1 precision and recall respectively) indicates that, regardless of having similar visual processing capabilities, LXMERT's design choices were much more optimized for cross-modality modeling.

On the one hand, LXMERT is trained on $\mathbf{9.1}$M text-image pairs, which almost *doubles* the $\mathbf{3.3}$M used during ViLBERT's pre-training. Moreover, the combination of several datasets with distinct characteristics for the creation of this model's pre-training corpus offers much better generalisation capabilities than only using one large collection of samples. On the other hand LXMERT's authors included additional pre-training tasks which offered a major boost in cross-modal understanding. Specifically *VqA*, which relied on the *overall VL representation of the entire input* (also used for this retrieval task) was crucial in the process of teaching the architecture how to extract the most useful information from both modalities.

ALBEF. The best-performing architecture in the *Image Classification* task did not manage to achieve the highest retrieval scores, regardless of its curated visual processing capabilities. Moreover, it only outperforms LXMERT by a small margin when compared to section 5.1 results.

This presents as a completely unexpected behaviour, since ALBEF's architecture is the most optimized in terms of uni-modal processing, leveraging *BERT's* linguistic capabilities for the encoding of text while making use of pre-trained *ViT* weights in its visual stream. Furthermore, ALBEF's pre-training corpus is the largest out of the selected models, with $\mathbf{14}$M text-image pairs, which should grant it an enormous advantage over its competitors.

According to these factors, ALBEF's performance in this multi-modal task is thought to be determined by the following two reasons: starting with the reason behind its low margin with respect to the next-best two-stream model, LXMERT is pre-trained on two tasks operating over

the *"overall VL representation of both streams"*, which is also used as input for the retrieval head in this benchmark. This forced the architecture to learn how to create a more effective mean pooled representation of the input than the one extracted by ALBEF, which only uses it for the task of *multi-modal alignment prediction*. On the other hand, its *decoder-like* modality-fusion mechanism, as described in section 3.2.5, appears to be a detrimental choice when compared to other strategies, where both streams are able to "attend" to each other.

ViLT. After ranking second in the Image classification task, where ALBEF managed to score $\mathbf{5.11\%}$ and $\mathbf{5.36\%}$ higher in terms of test and validation accuracy respectively, this architecture was able to reach first place in *retrieval* with at least $\mathbf{3\%}$ margin in precision scores. This outstanding result is determined by the following factors.

Firstly, ViLT's inherent visual abilities granted it an advantage over its BERT-based competitors. This indicates that ViT-based architectures pre-trained on linguistic-focused task such as *Masked language modelling* outperform BERT-based models which included visual-focused tasks such as *Masked multi-modal learning with ROI feature regression* in their pre-training.

On the other hand, the application of the optimal transport strategy via *word-patch alignment* did also grant this approach a performance boost with respect to architectures which used *contrastive learning* as their representation alignment method.

Finally, ViLT's modality fusion strategy clearly outperforms ALBEF's, due to its ability to allow both input types to attend to each other.

In light of the above, the key hypotheses developed according to the results obtained are the following:

- **Architecture related (Base Models):** ViT-based architectures are able to outperform BERT-based models in Vision-language tasks because, for a VL transformer it is easier to obtain linguistic understanding via pre-training objectives such as *Masked Language Modelling* than it is to learn visual processing capabilities when being trained on tasks such as *Masked Multi-Modal learning with ROI feature regression*. This makes ViT-based architectures, with weights initialised from an extensive visual-centered pre-training, considerably more skilled than its competitors.

- **Architecture related (Modality fusion strategy):** Decoder-like modality-fusion strategies, like the one present in ALBEF, have proven to be more detrimental in terms of performance than any other design choice in cross-modal tasks. The reason behind this behaviour is considered to be due to the fact that only one of the modalities (the linguistic in this case) is able to attend to the other.

- **Learning-related (VL Pre-training Objectives):** The inclusion of specific additional pre-training tasks, such as *VqA*, offer a massive performance boost since they force the model to learn how to better extract a useful overall representation of the entire input, which is in turn used in many other downstream tasks e.g. *multi-modal retrieval*, *visual entailment* or *visual commonsense reasoning*

6 CONCLUSIONS

This dissertation focused on investigating the impact that different design choices have on the performance of State-of-the-art VL models. Specifically, 3 classes of design choices were considered: (i) architecture-related, (ii) input feature-related, and (iii) learning related. This was achieved by performing detailed experimental comparison of a representative selection of VL methods developed within the field.

For the purpose of this investigation, the available literature published from 2019 onwards was reviewed, in order to identify and select a representative subset of 5 VL models which covered the design choice spectrum to the largest extent possible. The subset of approaches selected comprised of the following models: ViLBERT, LXMERT, Oscar, ViLT and ALBEF. Subsequently, the selected architectures where thoroughly analysed, enabling an in-depth comparison, grounded on a solid understanding of their structures and methods of work.

Thereafter, as part of the detailed benchmarking process, said models were implemented, adapted, fine-tuned and tested on two different tasks: Image classification and Text to Image retrieval. The dataset utilized for this was Places365-standard, a subset of MIT's Places2 database.

Based on our architectural analysis and the results obtained, the following observations concerning the different design choice classes were made:

Input feature related (Visual Features). The architectures which utilized non-overlapping flattened image patches as their visual inputs outperformed those which made use of the more computationally complex R-CNN ROI features.

Architecture related (Base Models). ViT-based architectures were able to outperform BERT-based approaches in both visual and VL tasks. In the case of purely visual tasks, ViT-based architectures have inherently superior image-processing capabilities, as they are initialised from weights pre-trained on large-scale CV datasets such as ImageNet, which grants them a tremendous advantage. In terms of VL objectives, it was found that said models are able to obtain the necessary linguistic understanding via pre-training objectives such as *Masked language modelling*, while it is

substantially harder for BERT-based architectures to learn visual processing capabilities via tasks such as *Masked multi-modal learning*.

Architecture related (Modality Fusion Strategy). The results of this investigation did not provide clear indications to state which modality fusion strategy is more optimal i.e. two-stream vs single-stream. However, it was found that, int the case of ALBEF, its decoder-like fusion mechanism, where only one modality (linguistic in this case) was able to ”attend” to the other, was a detrimental factor for its performance.

Learning related (Pre-training Objectives). *Masked multi-modal learning* and *Multi-modal alignment prediction* were found to be essential in the pre-training of VL architectures. The first one provided a mixture of linguistic and visual understanding capabilities particularly beneficial for both BERT and ViT based architectures. Furthermore, the regression of masked visual features granted a performance boost to the models which applied it. On the other hand, *Multi-modal alignment prediction* did not only force the models to adequately understand the relationship between modalities, but also to optimally condense the input information into a mean-pooled representation, which can then be used for various downstream tasks. The addition of supplementary pre-training tasks which operate over this mean-pooled representation, such as *VqA*, also offered improvements in performance.

Additionally, the selection of Image classification as one of the benchmarks allowed for the performance comparison between the selected models and other non-linguistic methods, such as Convolutional Neural Networks. This provides more information towards debunking or confirming the hypothesis which states that language-aligned visual representations can improve the current performance in purely-visual tasks. For this purpose, the metrics achieved by 4 CNNs trained and tested as part of Places365 original paper were extracted.

According to the results obtained, no visuo-linguistic approach was able to outperform the CNNs presented. However, major discrepancies between their training set-ups, in addition to the low margin in between their results may indicate that, the selected architectures could surpass the non-linguistic methods if the bias in their training was eliminated. Re-training all networks within a common paradigm enabling fair comparisons was considered but unfortunately it was not feasible within the time-frame and resources of the project.

6.1 Evaluation

This project has concluded with a positive outcome, achieving all its objectives.

To begin with, the latest literature regarding Vision an Language research was reviewed and selected models investigated to cluster approaches based on their distinctive features. The Background Theory and Literature review chapter summarises the state-of-the art and identifies the most prominent models, tasks and datasets, setting the scene for the design our study.

We then describe the selection process of the 5 VL architectures to be benchmarked as part of this investigation. Furthermore, the Methodology and Data chapter provides a thorough comparative analysis of selected architectures, fulfilling the second objective.

The selected architectures are fine-tuned and evaluated on two different tasks, Image classification and Text to Image retrieval, using Places365 as the dataset of choice. The tasks were designed to highlight the strengths and weaknesses of the models in joint vision-language tasks and in purely visual tasks. An explanation of the entire experimental set-up and detailed experimental results are included in this report, fulfilling the third objective.

Finally, the last objective required of the analysis of the results obtained. This was accomplished and documented as part of the final technical chapter, which attempts to give sense to the models' performance with respect to how they were developed and pre-trained. A new insight on the most promising models and good design choices was developed and is the most important output of this study.

6.2 Future Work

We believe that the work carried out for this dissertation forms a solid base in the investigation of SoTA VL models. However, we have also identified several areas where further work is required to expand our knowledge about VL architectures. Some ideas are as follows:

The selected subset of SoTA methods could be extended, in order to include more directly-comparable architectures i.e. direct successors such as the included ViLBERT and LXMERT.

The training of the selected networks was constraint by time and GPU resources available. However, in order to test the full capabilities of the models, they should be trained in a more flexible manner: end–to-end training (i.e. without frozen weights) with longer epochs and on

larger/more diverse datasets . Furthermore, an appropriate comparison with non-linguistic methods could also be performed by training them under a common set-up.

In order to further understand the effect that different model-specific components have on the final performance, appropriate ablation studies could be carried out, where these modules are removed from the architectures prior to the benchmarking process.

Finally, the knowledge generated as part of this dissertation could be used towards the creation of a VL architecture containing the most promising design choices identified. This new model could then also be compared in order to re-affirm the hypothesis made in this project.

BIBLIOGRAPHY

[1] What is the learing rate schedule. https://peltarion.com/knowledge-center/modeling-view/run-a-model/optimization-principles-(in-deep-learning)/learning-rate-schedule.

[2] Image Recognition and Object Detection : Part 1 — LearnOpenCV #, Nov. 2016.

[3] Agrawal, A. , Lu, J. , Antol, S. , Mitchell, M. , Zitnick, C. L. , Batra, D. , and Parikh, D. . VQA: Visual Question Answering, Oct. 2016.

[4] Agrawal, A. , Lu, J. , Antol, S. , Mitchell, M. , Zitnick, C. L. , Batra, D. , and Parikh, D. . VQA: Visual Question Answering, Oct. 2016.

[5] Agrawal, N. . Understanding Attention Mechanism: Natural Language Processing, Jan. 2020.

[6] Anderson, P. , He, X. , Buehler, C. , Teney, D. , Johnson, M. , Gould, S. , and Zhang, L. . Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering, Mar. 2018.

[7] Ba, J. L. , Kiros, J. R. , and Hinton, G. E. . Layer Normalization, July 2016.

[8] Bahdanau, D. , Cho, K. , and Bengio, Y. . Neural Machine Translation by Jointly Learning to Align and Translate, May 2016.

[9] Bay, H. , Tuytelaars, T. , and Van Gool, L. . SURF: Speeded Up Robust Features. In Leonardis, A. , Bischof, H. , and Pinz, A. , editors, *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, pages 404–417, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-33833-8. doi: 10.1007/11744023_32.

[10] Brown, T. B. , Mann, B. , Ryder, N. , Subbiah, M. , Kaplan, J. , Dhariwal, P. , Neelakantan, A. , Shyam, P. , Sastry, G. , Askell, A. , Agarwal, S. , Herbert-Voss, A. , Krueger, G. , Henighan, T. , Child, R. , Ramesh, A. , Ziegler, D. M. , Wu, J. , Winter, C. , Hesse, C. , Chen, M. , Sigler, E. , Litwin, M. , Gray, S. , Chess, B. , Clark, J. , Berner, C. , McCandlish, S. , Radford, A. , Sutskever, I. , and Amodei, D. . Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020.

[11] Changpinyo, S. , Sharma, P. , Ding, N. , and Soricut, R. . Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts, Mar. 2021.

[12] Cho, K. , van Merrienboer, B. , Gulcehre, C. , Bahdanau, D. , Bougares, F. , Schwenk, H. , and Bengio, Y. . Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. June 2014. doi: 10.48550/arXiv.1406.1078.

[13] Dalal, N. and Triggs, B. . Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. doi: 10.1109/CVPR.2005.177.

[14] Dang, N. C. , Moreno-García, M. N. , and De la Prieta, F. . Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics*, 9(3):483, Mar. 2020. ISSN 2079-9292. doi: 10.3390/electronics9030483.

[15] Devlin, J. , Chang, M.-W. , Lee, K. , and Toutanova, K. . BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019.

[16] Doshi, K. . Transformers Explained Visually (Part 1): Overview of Functionality. https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452, June 2021.

[17] Dosovitskiy, A. , Beyer, L. , Kolesnikov, A. , Weissenborn, D. , Zhai, X. , Unterthiner, T. , Dehghani, M. , Minderer, M. , Heigold, G. , Gelly, S. , Uszkoreit, J. , and Houlsby, N. . An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*, June 2021.

[18] Fukui, A. , Park, D. H. , Yang, D. , Rohrbach, A. , Darrell, T. , and Rohrbach, M. . Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1044.

[19] Ganegedara, T. . *Natural Language Processing with TensorFlow: Teach Language to Machines Using Python's Deep Learning Library*. Packt Publishing Ltd, May 2018. ISBN 978-1-78847-775-8.

[20] Geetha, G. , Kirthigadevi, T. , Ponsam, G. , Karthik, T. , and Safa, M. . Image Captioning Using Deep Convolutional Neural Networks (CNNs). *Journal of Physics: Conference Series*,

1712(1):012015, Dec. 2020. ISSN 1742-6588, 1742-6596. doi: 10.1088/1742-6596/1712/1/012015.

[21] Girshick, R. , Donahue, J. , Darrell, T. , and Malik, J. . Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv:1311.2524 [cs]*, Oct. 2014.

[22] He, K. , Zhang, X. , Ren, S. , and Sun, J. . Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, Dec. 2015.

[23] Hendrycks, D. and Gimpel, K. . Gaussian Error Linear Units (GELUs), July 2020.

[24] Hochreiter, S. and Schmidhuber, J. . Long Short-term Memory. *Neural computation*, 9: 1735–80, Dec. 1997. doi: 10.1162/neco.1997.9.8.1735.

[25] Huang, Z. , Zeng, Z. , Liu, B. , Fu, D. , and Fu, J. . Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers, June 2020.

[26] Hudson, D. A. and Manning, C. D. . GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00686.

[27] Ibanez, D. . Encoder-Decoder Models for Natural Language Processing — Baeldung on Computer Science. https://www.baeldung.com/cs/nlp-encoder-decoder-models, Jan. 2021.

[28] Johnson, J. , Karpathy, A. , and Fei-Fei, L. . DenseCap: Fully Convolutional Localization Networks for Dense Captioning, Nov. 2015.

[29] Kafle, K. and Kanan, C. . Answer-Type Prediction for Visual Question Answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4976–4984, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.538.

[30] Kim, W. , Son, B. , and Kim, I. . ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. *arXiv:2102.03334 [cs, stat]*, June 2021.

[31] Krishna, R. , Zhu, Y. , Groth, O. , Johnson, J. , Hata, K. , Kravitz, J. , Chen, S. , Kalantidis, Y. , Li, L.-J. , Shamma, D. A. , Bernstein, M. S. , and Fei-Fei, L. . Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal*

of Computer Vision, 123(1):32–73, May 2017. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-016-0981-7.

[32] Krizhevsky, A. , Sutskever, I. , and Hinton, G. E. . ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[33] LeCun, Y. , Bottou, L. , Bengio, Y. , and Ha, P. . Gradient-Based Learning Applied to Document Recognition. page 46, 1998.

[34] Li, F. , Zhang, H. , Zhang, Y.-F. , Liu, S. , Guo, J. , Ni, L. M. , Zhang, P. , and Zhang, L. . Vision-Language Intelligence: Tasks, Representation Learning, and Large Models. *arXiv:2203.01922 [cs]*, Mar. 2022.

[35] Li, J. , Selvaraju, R. R. , Gotmare, A. D. , Joty, S. , Xiong, C. , and Hoi, S. . Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. *arXiv:2107.07651 [cs]*, Oct. 2021.

[36] Li, X. , Yin, X. , Li, C. , Zhang, P. , Hu, X. , Zhang, L. , Wang, L. , Hu, H. , Dong, L. , Wei, F. , Choi, Y. , and Gao, J. . Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. *arXiv:2004.06165 [cs]*, July 2020.

[37] Lin, T.-Y. , Maire, M. , Belongie, S. , Bourdev, L. , Girshick, R. , Hays, J. , Perona, P. , Ramanan, D. , Zitnick, C. L. , and Dollár, P. . Microsoft COCO: Common Objects in Context, Feb. 2015.

[38] Long, S. , Cao, F. , Han, S. C. , and Yang, H. . Vision-and-Language Pretrained Models: A Survey. *arXiv:2204.07356 [cs]*, Apr. 2022.

[39] Loshchilov, I. and Hutter, F. . Decoupled Weight Decay Regularization, Jan. 2019.

[40] Lu, J. , Batra, D. , Parikh, D. , and Lee, S. . ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *arXiv:1908.02265 [cs]*, Aug. 2019.

[41] Marr, D. . *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, Cambridge, Mass, 2010. ISBN 978-0-262-51462-0.

[42] Mezaris, V. , Kompatsiaris, I. , and Strintzis, M. . An ontology approach to object-based image retrieval. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 3, pages II–511–14, Barcelona, Spain, 2003. IEEE. ISBN 978-0-7803-7750-9. doi: 10.1109/ICIP.2003.1246729.

[43] Mikolov, T. , Chen, K. , Corrado, G. , and Dean, J. . Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, Sept. 2013.

[44] Mogadala, A. , Kalimuthu, M. , and Klakow, D. . Trends in Integration of Vision and Language Research: A Survey of Tasks, Datasets, and Methods. *Journal of Artificial Intelligence Research*, 71:1183–1317, Aug. 2021. ISSN 1076-9757. doi: 10.1613/jair.1.11688.

[45] Moresi, M. . $\frac{3}{4}$ Applications in dialogue systems $\frac{3}{4}$ Conclusion. page 46.

[46] O'Mahony, N. , Campbell, S. , Carvalho, A. , Harapanahalli, S. , Hernandez, G. V. , Krpalkova, L. , Riordan, D. , and Walsh, J. . Deep Learning vs. Traditional Computer Vision. In Arai, K. and Kapoor, S. , editors, *Advances in Computer Vision*, volume 943, pages 128–144. Springer International Publishing, Cham, 2020. ISBN 978-3-030-17794-2 978-3-030-17795-9. doi: 10.1007/978-3-030-17795-9_10.

[47] Ordonez, V. , Kulkarni, G. , and Berg, T. . Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[48] Prabhu. Understanding of Convolutional Neural Network (CNN) — Deep Learning, Nov. 2019.

[49] Radford, A. , Narasimhan, K. , Salimans, T. , and Sutskever, I. . Improving Language Understanding by Generative Pre-Training. page 12, .

[50] Radford, A. , Wu, J. , Child, R. , Luan, D. , Amodei, D. , and Sutskever, I. . Language Models are Unsupervised Multitask Learners. page 24, .

[51] Ren, S. , He, K. , Girshick, R. , and Sun, J. . Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. page 9.

[52] Sharma, P. , Ding, N. , Goodman, S. , and Soricut, R. . Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238.

[53] Sherstinsky, A. . Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, 404:132306, Mar. 2020. ISSN 01672789. doi: 10.1016/j.physd.2019.132306.

[54] Simonyan, K. and Zisserman, A. . Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Apr. 2015.

[55] Su, W. , Zhu, X. , Cao, Y. , Li, B. , Lu, L. , Wei, F. , and Dai, J. . VL-BERT: Pre-training of Generic Visual-Linguistic Representations. *arXiv:1908.08530 [cs]*, Feb. 2020.

[56] Tan, H. and Bansal, M. . LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *arXiv:1908.07490 [cs]*, Dec. 2019.

[57] Tan, M. and Le, Q. V. . EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, Sept. 2020.

[58] Vaswani, A. , Shazeer, N. , Parmar, N. , Uszkoreit, J. , Jones, L. , Gomez, A. N. , Kaiser, L. , and Polosukhin, I. . Attention Is All You Need. *arXiv:1706.03762 [cs]*, Dec. 2017.

[59] Vinyals, O. , Toshev, A. , Bengio, S. , and Erhan, D. . Show and Tell: A Neural Image Caption Generator, Apr. 2015.

[60] Xie, Y. , Wang, X. , Wang, R. , and Zha, H. . A Fast Proximal Point Method for Computing Exact Wasserstein Distance. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 433–453. PMLR, Aug. 2020.

[61] Young, P. , Lai, A. , Hodosh, M. , and Hockenmaier, J. . From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166.

[62] Zhou, B. , Lapedriza, A. , Khosla, A. , Oliva, A. , and Torralba, A. . Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, June 2018. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2017.2723009.