A Study on Vision-Language Models

Hugo Arsénio, Matteo Mello Grand, Riccardo Bertamini

5 November 2024

Contents

1 Introduction **2**
1.1 Background 2

2 VLM Foundations **3**
2.1 Pre-training frameworks 3
2.2 Pre-training Architectures 3
2.3 The Objectives 4

3 VLM Pre-Training **4**

4 VLM Transfer Learning **5**

5 VLM Knowledge Distillation **5**

6 Performance Comparison **6**

7 Future Prospects **6**
7.1 VLM Pre-Training 6
7.2 VLM transfer learning 6
7.3 VLM knowledge distillation 6

8 Conclusion **7**

9 Practical **7**

1 Introduction

1.1 Background

Vision tasks are one of the domains of application of Artificial Intelligence; they include different types of problems such as image recognition, object detection, and semantic segmentation. In this article, we are going to analyze how vision tasks can be solved using Vision Language Models (VLMs).

Traditionally, these tasks were solved using standard machine learning algorithms such as Support Vector Machines (SVMs). This approach, however, required manual feature engineering, which is time-consuming and not scalable for complex problems.

Luckily, once deep learning was invented, new deep learning-based solutions to vision tasks were proposed. An example is Convolutional Neural Networks (CNNs). This new approach does not require manual feature engineering since the neural network itself learns features from the images by adjusting its parameters. However, the application of deep learning has two main problems:

1. It requires large-scale task-specific crowd-labelled data.

2. It has slow convergence.

To solve these two problems, the research community introduced a new paradigm based on pre-training.

In this new architecture, a first training of a Deep Neural Network (DNN) is done on generic data, and then a second training called *Fine Tuning* is performed using task-specific data. This new approach has a faster convergence. Moreover, if the pre-training is performed using self-supervised machine learning algorithms, it is possible to use unlabelled data rather than labelled ones.

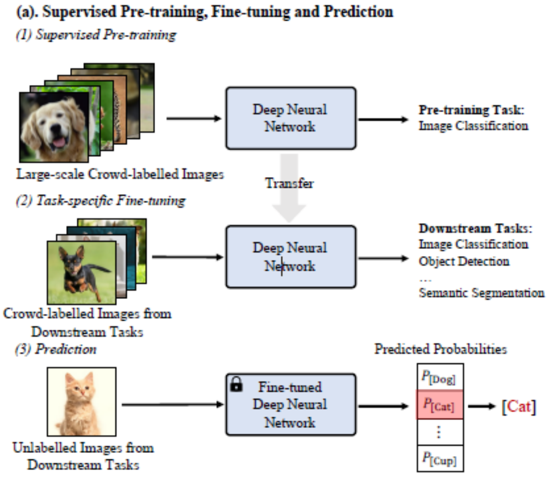

(a). Supervised Pre-training, Fine-tuning and Prediction
(1) Supervised Pre-training
Deep Neural Network
Pre-training Task:
Image Classification
Large-scale Crowd-labelled Images
Transfer
(2) Task-specific Fine-tuning
Deep Neural Network
Downstream Tasks:
Image Classification
Object Detection
...
Semantic Segmentation
Crowd-labelled Images from Downstream Tasks
(3) Prediction
Predicted Probabilities
Fine-tuned Deep Neural Network
$P_{[Dog]}$
$P_{[Cat]}$
⋮
$P_{[Cup]}$
[Cat]
Unlabelled Images from Downstream Tasks

Figure 1: Example of pre-training approach.

However, this approach has still a problem: the necessity of large amounts of data and specific-task data. Vision Language Models (VLMs) can solve these problems.

2 VLM Foundations

Vision Language Models are based on a *Pre-training* → *Zero-shot Prediction* architecture. They feature an initial pre-training phase, followed by direct prediction without additional fine-tuning. The main difference from previous pre-training methods is that VLMs use paired text-image data during pre-training. By leveraging both images and text, VLMs can extract richer insights from the data, allowing the model to generalize better without fine-tuning.

Additionally, it is easy to find the data: indeed, text-image paired data are easily findable online on the web (for example, images and their captions).

2.1 Pre-training frameworks

There are three main frameworks that can be used for pre-training:

1. **Two Tower VLM**

   It consists of two deep neural networks, respectively for images and text. They are trained separately but according to the same set of objectives. Therefore, the DNN for images is influenced by text, and the DNN for text is influenced by images.

2. **Two leg VLM**

   his is similar to the two-tower VLM but introduces a multi-modal fusion layer.

3. **One Tower VLM**

   It is the simplest framework. It consists of just a single Deep Neural Network which is common to both images and text. The DNN is trained according to a set of objectives that depend on both text and images.

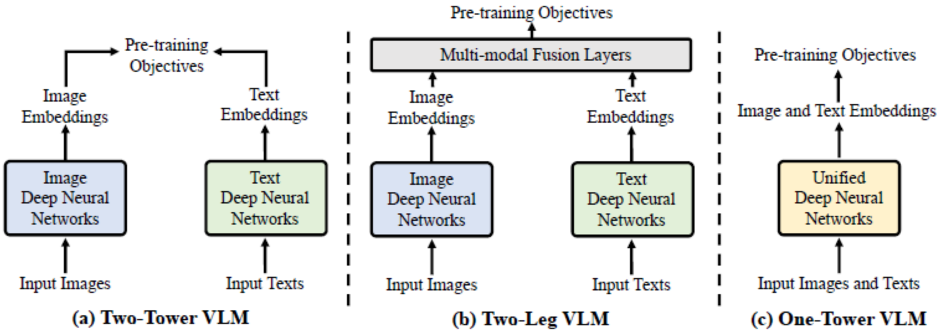


Figure 2: VLM frameworks

2.2 Pre-training Architectures

Once a framework is chosen, the architecture of the DNNs involved in the pre-training needs to be determined. For images, it is common to use Convolutional Neural Networks or transformers. For text, it is very common to use a standard transformer architecture.

2.3 The Objectives

There are three main types of objectives that can be used to train the DNNs. Let's analyze them one by one:

1. **Contrastive Objectives**

   Given an embedding space, a contrastive objective brings paired text and images closer to each other, while it pushes apart images and text that are not paired.

2. **Generative Objectives**

   With generative objectives, the model generates data and checks whether it is correct or not. If it is not correct, then the model adjusts its parameters accordingly.

3. **Alignment Objectives**

   Alignment objectives teach the model to link images and text. Particularly, there are two types of alignment objectives:

   (a) **Image-Text Matching**: It learns to match images and text together.

   (b) **Region-Word Matching**: It models correlations between sections of an image and substrings of the text.

3 VLM Pre-Training

Vision-Language Model (VLM) pre-training is a pivotal step in learning image-text correlations. This phase involves the following key aspects:

- **Data Utilization:** Relies on web-scale datasets containing billions of diverse image-text pairs, enabling models to generalize across varied contexts.
- **Pre-Training Objectives:**
  - **Contrastive Learning:** Aligns image and text embeddings by pulling paired data closer and pushing unrelated pairs further apart in a shared embedding space, as used in models like CLIP.
  - **Generative Modeling:** Encourages models to reconstruct masked image patches or text tokens, reinforcing contextual understanding.
  - **Alignment Strategies:** Enhances the semantic relationship between images and text for downstream applications.
- **Architectures:** Leverages advanced frameworks like Transformers for encoding, capturing nuanced details across modalities.
- **Zero-Shot Predictions:** Enables models to apply knowledge to unseen tasks without task-specific fine-tuning, showcasing their versatility.

This structured pre-training process ensures that VLMs are adaptable to downstream tasks like classification, segmentation, and retrieval, with robust multimodal understanding.

4 VLM Transfer Learning

VLM transfer learning focuses on adapting pre-trained models to specific tasks. The core strategies include:

Prompt Tuning: Modifies input prompts to help the model interpret domain-specific contexts without retraining the architecture. Particularly effective in:

- **Few-Shot Scenarios:** Achieves high performance with limited labeled data.
- **Unified Prompt Tuning:** Combines textual and visual cues for enhanced task performance.

Visual Adaptation: Integrates feature adapters into VLM pipelines to refine task-relevant image representations. For instance:

- **Visual Prompt Tuning:** Adjusts specific layers to focus on domain-specific visual features.

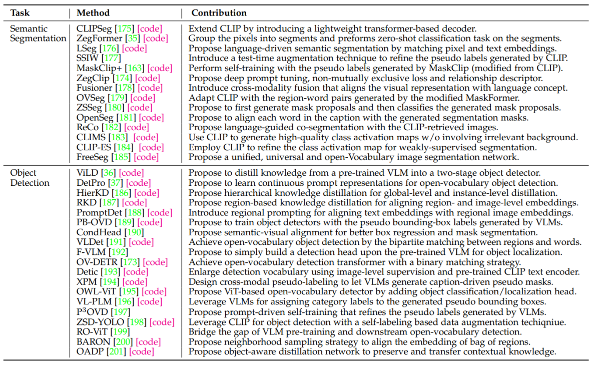
Efficiency: Transfer learning avoids extensive retraining by building on foundational knowledge from pre-training, making it resource-efficient while ensuring high accuracy.

These techniques unlock applications such as dense predictions and multi-label classification, demonstrating the adaptability and efficiency of VLM transfer learning.

5 VLM Knowledge Distillation

VLMs capture generalizable knowledge covering a wide range of visual and text concepts, and VLM knowledge distillation takes the most important part of the knowledge gathered. They do this using task-specific models. Most VLM knowledge distillation methods focus on transferring image-level knowledge to region or pixel-level tasks. To understand how it manages complex dense predictions we can use two different examples:

- Object detection. The aim is to better align image-level and object-level representations. It does this using two methods:
  - Open vocabulary object detection: aims to detect objects described by arbitrary texts, so objects of any categories beyond the base classes.
  - VLM distillation via prompt learning (PL). PL allows partial parts of prompts to be trainable, and improves performance.
- Semantic segmentation. Tackles mismatches between image-level and pixel-level representations. There are two cases:
  - Open-vocabulary of segmentation models, which aims to segment pixels described by arbitrary texts.
  - Knowledge distillation for weakly-supervised semantic segmentation, aims to leverage both VLMs and weak supervision for semantic segmentation.



Task	Method	Contribution
Semantic Segmentation	CLIPSeg [175] [code]	Extend CLIP by introducing a lightweight transformer-based decoder.
	ZegFormer [35] [code]	Group the pixels into segments and preforms zero-shot classification task on the segments.
	LSeg [176] [code]	Propose language-driven semantic segmentation by matching pixel and text embeddings.
	SSIW [177]	Introduce a test-time augmentation technique to refine the pseudo labels generated by CLIP.
	MaskClip+ [163] [code]	Perform self-training with the pseudo labels generated by MaskClip (modified from CLIP).
	ZegClip [174] [code]	Propose deep prompt tuning, non-mutually exclusive loss and relationship descriptor.
	Fusioner [178] [code]	Introduce cross-modality fusion that aligns the visual representation with language concept.
	OVSeg [179] [code]	Adapt CLIP with the region-word pairs generated by the modified MaskFormer.
	ZSSeg [180] [code]	Propose to first generate mask proposals and then classifies the generated mask proposals.
	OpenSeg [181] [code]	Propose to align each word in the caption with the generated segmentation masks.
	ReCo [182] [code]	Propose language-guided co-segmentation with the CLIP-retrieved images.
	CLIMS [183] [code]	Use CLIP to generate high-quality class activation maps w/o involving irrelevant background.
	CLIP-ES [184] [code]	Employ CLIP to refine the class activation map for weakly-supervised segmentation.
	FreeSeg [185] [code]	Propose a unified, universal and open-Vocabulary image segmentation network.
Object Detection	ViLD [36] [code]	Propose to distill knowledge from a pre-trained VLM into a two-stage object detector.
	DetPro [37] [code]	Propose to learn continuous prompt representations for open-vocabulary object detection.
	HierKD [186] [code]	Propose hierarchical knowledge distillation for global-level and instance-level distillation.
	RKD [187] [code]	Propose region-based knowledge distillation for aligning region- and image-level embeddings.
	PromptDet [188] [code]	Introduce regional prompting for aligning text embeddings with regional image embeddings.
	PB-OVD [189] [code]	Propose to train object detectors with the pseudo bounding-box labels generated by VLMs.
	CondHead [190]	Propose semantic-visual alignment for better box regression and mask segmentation.
	VLDet [191] [code]	Achieve open-vocabulary object detection by the bipartite matching between regions and words.
	F-VLM [192]	Propose to simply build a detection head upon the pre-trained VLM for object localization.
	OV-DETR [173] [code]	Achieve open-vocabulary detection transformer with a binary matching strategy.
	Detic [193] [code]	Enlarge detection vocabulary using image-level supervision and pre-trained CLIP text encoder.
	XPM [194] [code]	Design cross-modal pseudo-labeling to let VLMs generate caption-driven pseudo masks.
	OWL-ViT [195] [code]	Propose ViT-based open-vocabulary detector by adding object classification/localization head.
	VL-PLM [196] [code]	Leverage VLMs for assigning category labels to the generated pseudo bounding boxes.
	P^{3}OVD [197]	Propose prompt-driven self-training that refines the pseudo labels generated by VLMs.
	ZSD-YOLO [198] [code]	Leverage CLIP for object detection with a self-labeling based data augmentation techiqniue.
	RO-ViT [199]	Bridge the gap of VLM pre-training and downstream open-vocabulary detection.
	BARON [200] [code]	Propose neighborhood sampling strategy to align the embedding of bag of regions.
	OADP [201] [code]	Propose object-aware distillation network to preserve and transfer contextual knowledge.

Figure 3: Advancements in VLMs.

6 Performance Comparison

VLM pre-training achieves remarkable zero-shot prediction on a wide range of image classification tasks due to its well-designed pre-training objectives, but its development for dense visual recognition tasks lags far behind. VLM transfers have made remarkable progress across image classification datasets; however, supervised or few-shot supervised transfer requires labeled images, while unsupervised transfer has been neglected. VLM knowledge distillation is much harder to judge since many studies adopt different task-specific backbones and benchmarking in a fair way is much harder.

7 Future Prospects

Some future studies regarding VLMs could be:

7.1 VLM Pre-Training

- Data-efficient VLM. VLMs need large-scale training data and intensive computations, making its sustainability a big concern, and training VLMs using limited image-text data could mitigate the issue.

- Pre-training VLMs with LLMs. Employ LLMs to augment texts in the raw image-text pairs, providing richer language knowledge, helping better learn the vision-language correlation.

7.2 VLM transfer learning

- Unsupervised VLM transfer. Which would enable VLM transfers to have a much lower risk of overfitting.

7.3 VLM knowledge distillation

- Distilling knowledge from multiple VLMs, harvesting their synergistic effect by coordinating the knowledge distillation from multiple VLMs.

8 Conclusion

VLMs enable effective usage of web data and allow zero-shot predictions without task-specific fine-tuning, which is simple to implement yet has achieved great success on a wide range of recognition tasks.

9 Practical

This study demonstrates a practical application of Vision-Language Models (VLMs) by leveraging OpenAI's CLIP model to pair images with captions effectively. The process is outlined as follows:

- **Model Initialization and Preprocessing:**
  - The CLIP model, pre-trained on large-scale datasets of image-text pairs, is initialized.
  - Images are preprocessed from a specified folder to prepare them for evaluation.
- **Caption Pairing:**
  - The model scores each image-caption pair based on semantic similarity.
  - Predefined captions are used to match images effectively, demonstrating the strength of transfer learning.
- **Real-World Efficiency:**
  - By leveraging pre-trained capabilities, the model operates without additional training, making it highly efficient for applications with constrained data.
  - Transfer learning ensures high performance in domain-specific tasks, such as pairing visual humor with textual expressions.
- **Performance Enhancement:**
  - Knowledge distillation is proposed as a refinement method to train smaller, task-specific models.
  - This approach improves deployment efficiency while retaining the interpretability of CLIP's decision-making.
- **Output and Visualization:**
  - The script iterates through the images and captions, selecting the best match for each image.
  - Captions are overlayed on the images to create accurate and humorous memes.

This hands-on implementation exemplifies the integration of advanced VLM capabilities into creative, everyday applications, showcasing their potential in both technical and artistic domains.