ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

BACHELOR SEMESTER PROJECT - REPORT

# EXPLORING LARGE VISION-LANGUAGE MODELS FOR HISTORICAL IMAGE CLASSIFICATION

INES BOUCHAMA

---

4th January 2024

---

PROJECT SUPERVISORS:
Dr. Emanuela Boros
Dr. Maud Ehrmann

## ABSTRACT

This semester project delves into the assessment of vision models like CLIP and Flamingo for the **automatic classification and captioning of historical images**. It confronts the challenges related to the quality of historical images, complex layouts, and artefact noise. The project critically evaluates the models' effectiveness in **zero-shot and few-shot image classification**, as well as explores the multilingual capabilities of CLIP and the optimal prompting strategies for improved classification. Additionally, it investigates Flamingo's ability to generate accurate, consistent, and contextually appropriate **image captions**. The report is distinctive for its comprehensive exploration of **prompt engineering** and provides a nuanced quantitative and qualitative analysis of the models' performances, shedding light on the advancements and potential future directions in the field of automatic historical image classification and captioning.


# Contents

# 1 Introduction

The *impresso*[1] project – "*Media Monitoring of the Past*" carried out at the Digital Humanities Laboratory at EPFL is an interdisciplinary research project aiming for the datafication of a multilingual corpus of digitized historical newspapers. The project features a dataset of around 90 digitised historical newspapers containing approximately three million images. These images have no labels, and only 10% of them have a caption, two aspects that hinder their retrieval.

In the context of *impresso*, we focus on exploring new ways of automatically classifying and captioning historical images, with the aim of integrating these techniques into historical research workflows.

Image classification and image captioning play crucial roles in the digital preservation and accessibility of historical newspapers by enabling efficient organization and categorization of historical newspaper content. This facilitates easy searching and retrieval of specific information, articles, or images, greatly enhancing the accessibility of these resources to researchers, historians, and the general public. However, there are several challenges in automatic image captioning and classification:

- **Poor image quality**: Historical newspapers often suffer from fading, smudging, and physical damage, leading to poor image quality. This can hinder the accuracy of image recognition and captioning algorithms.
- **Complex layouts**: The varied and complex layouts, along with different font styles used in historical newspapers, can complicate text extraction and image classification.
- **Noise and artefacts**: Scanning artefacts, ink smudges, and paper quality issues introduce noise into the digitized images, making it difficult for algorithms to accurately identify and classify content.
- **Incomplete or missing captions**: Often, captions may be partially missing, illegible, or completely absent, necessitating the generation of entirely new captions based on the image content and surrounding context.

In this report, we explore several multimodal vision language models and offer insights on several research directions (Bisk et al. 2020):

1. *Are large vision models good at image classification?* – We evaluate zero-shot image type classification of the available dataset using the CLIP (Contrastive Language–Image Pretraining) (Radford et al. 2021), and Flamingo (Alayrac et al. 2022) in two settings, zero-shot and few-shot. We then compare with previous performances;
2. *What are the considerations for choosing a captioning model?* – We evaluate image captioning on the same dataset, by an explorative study of prompting Flamingo, a multimodal, vision-language model, known for its ability in generating captions in two settings, zero-shot and few-shot.

[1] `https://impresso-project.ch/`

# 2 Image Dataset

For our experiments, we use the *impresso image classification* dataset[1] which originally consisted of 7,915 images classified into ten different classes, as presented in Figure 2.1.

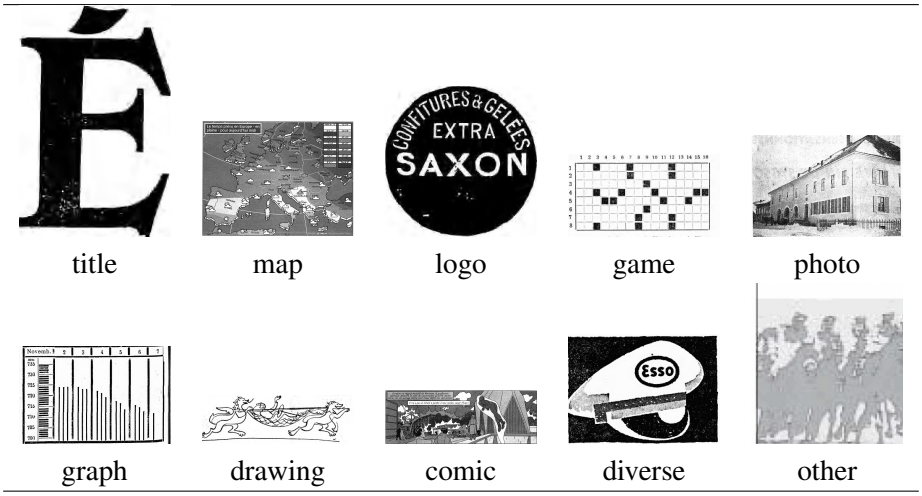



**TABLE 2.1**
Examples of images belonging to each class.

This dataset contains a train set and a test set. The test set was mainly used, except for few-shot classification, where the examples given to the model were extracted from the train set. The statistics of this set are presented in Table 2.2.

| **Class** | Title | Map | Logo | Game | Photo | Graph | Drawing | Comic | Diverse | Other |
|---|---|---|---|---|---|---|---|---|---|---|
| **Count** | 81 | 17 | 46 | 46 | 375 | 37 | 90 | 41 | 64 | 5 |

**TABLE 2.2**
Class names with corresponding counts in the test set.

After a few experiments, noticing that the classes "Diverse" and "Other" were too broad and overlapping with the other classes, we decided that it was not feasible to evaluate a classification task on them. The performance of the models achieved on those classes was not at all representative of the model's capabilities. They have thus been eliminated from the dataset used in this project.

[1]More specifically, we used collection under the *impresso_raw* folder from `https://github.com/impresso/impresso-image-classification`.

# 3 Image Classification

In this section, we present our experimental setup for the image classification task. We test two recent multimodal vision models, CLIP and Flamingo. We chose these two different models because, on one hand, CLIP is primarily a vision model that learns from text-image pairs to understand and categorize images in a human-like way, excelling in image recognition and association tasks. Flamingo, on the other hand, is a more holistic multimodal model that combines the capabilities of large language models with visual processing, making it adept at tasks that require an integrated understanding of both text and images.

## 3.1 CLIP

CLIP (Contrastive Language–Image Pre-training) developed by OpenAI, is a vision model that learns visual concepts from natural language supervision (Radford et al. 2021). It is trained on a large variety of internet text-image pairs. CLIP is designed to understand and classify images in a way that's more aligned with human-like understanding. It can recognize a broad range of visual concepts in images and associate them with natural language descriptions being trained with a contrastive learning approach, which involves learning to associate images with their corresponding textual descriptions. This means that it learns to understand images in the context of how they are described in text, which is a significant departure from traditional image recognition models that rely on labelled datasets. Since its release, CLIP has been used for powerful applications such as generating text-conditional images by combining it with diffusion models (Ramesh et al. 2022).

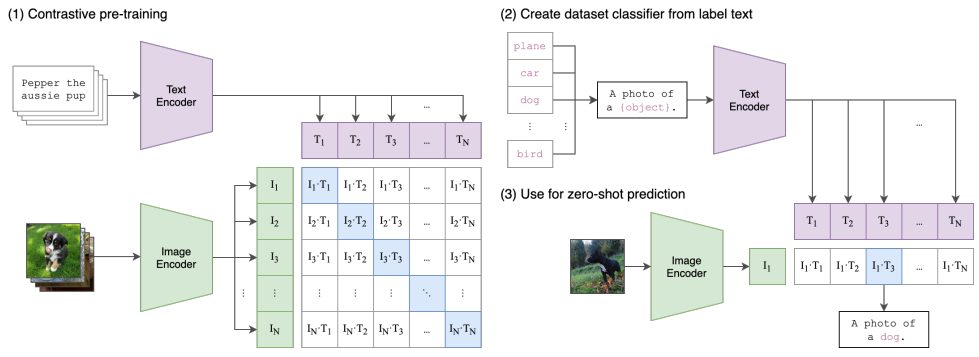



**FIGURE 3.1**
Architecture of CLIP from Radford et al. 2021.

As presented in Figure 3.1, CLIP works by taking textual prompts and images as inputs, which it encodes respectively. It then maps them to a space which enables to compute the distance between each text prompt and image prompt. For the zero-shot classification task, we want to pair the closest images and texts. CLIP outputs the probability that each image belongs to each class, making it relatively simple to extract the class with the highest probability for each image.

**HYPERPARAMETERS** For the implementation of CLIP, besides the original repository provided by OpenAI (*CLIP repository* 2022), we also took inspiration from (*Zero-shot Image Classification with OpenAI's CLIP* 2022; *Multi-modal ML with OpenAI's CLIP* 2022). We chose the pretrained CLIP model `clip-vit-base-patch32`[1] and it uses a ViT-B/32 Transformer architecture (Dosovitskiy et al. 2020) as an image encoder and uses a masked self-attention Transformer as a text encoder, with default hyperparameters.

| Index | Prompt | Example |
|---|---|---|
| | **Specification prompts** | |
| 0 | class name | "comic" |
| 1 | "an image of a" + class name | "an image of a comic" |
| 2 | noun phrase | "traditional comic" |
| 3 | adjective 1 + adjective 2 + class name | "entertaining traditional comic" |
| 4 | disjunction of noun phrases | "adventure comic or humour comic or superhero comic or war comic" |
| 5 | noun phrases separated by commas | "adventure comic, humour comic, superhero comic, war comic" |
| 6 | "This" + class name + verb + preposition | "This comic is a story told with successive images" |
| | **Paraphrasing prompts** | |
| 7 | class guidelines | "Comic or Satirical Drawing: Any humour or entertainment drawing. Could include text bubbles or text descriptions. Could be both one image or multiple to create a mini graphic novel." |
| 8 | class name + ":" + adapted Cambridge definition | "comic: a set of stories told in pictures with a small amount of writing" |
| 9 | adapted Cambridge definition | "a set of stories told in pictures with a small amount of writing" |
| 10 | class name + ":" + Cambridge definition | "comic: a set of stories told in pictures with a small amount of writing" |
| 11 | Cambridge definition | "a set of stories told in pictures with a small amount of writing" |
| 12 | ChatGPT analogy of the class to the prompt *Give me an analogy that captures the essence of a comic* | "A comic is like a rollercoaster ride for your imagination. Just as a rollercoaster takes you on a thrilling journey with its twists, turns, and unexpected drops." |

**TABLE 3.1**
Prompt variations for zero-shot classification using CLIP with examples on the class *comic*.

### 3.1.1 PROMPTING CLIP FOR ZERO-SHOT IMAGE CLASSIFICATION

We are particularly interested in determining how to best prompt the model to gain insights on its true potential and optimize its performance. To have a first glimpse at how CLIP behaves, we prompted it using only the eight class names. Then, we curated a list of prompts from simple to more complex, presented in Table 3.1. We distinguish specification prompts from paraphrasing prompts:

- **Specification prompts**: They focus on providing specific details and characteristics related to the

[1] `https://huggingface.co/openai/clip-vit-base-patch32`

target class. They aim to specify attributes, features, and categorizations associated with the given class.

- **Paraphrasing prompts**: They restate, define the target class in alternative ways. They aim to convey the essence or definition using different linguistic constructs, promoting a varied understanding of the same concept.

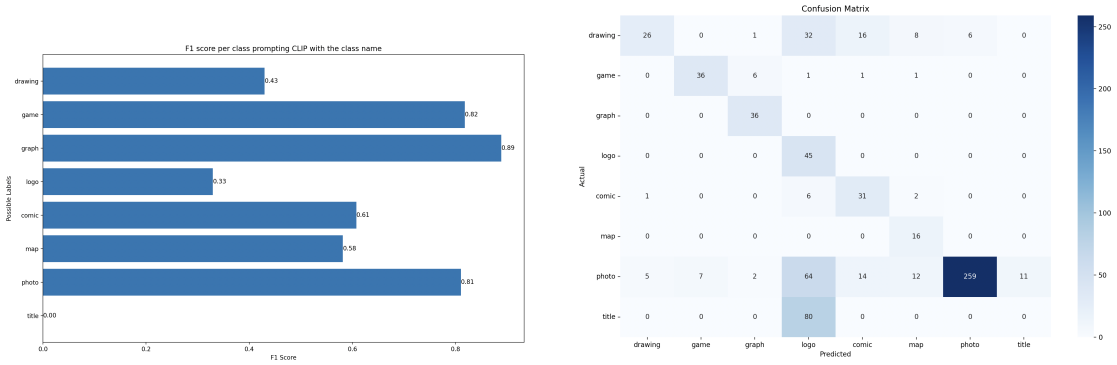
The prompt variations were carefully designed with the intention to answer the following questions:

1. How sensitive is CLIP to small changes in wording, connectors, punctuation and length of prompt?
2. Does CLIP understand nuances of words? Can it associate them to a general concept?
3. What is the prompt pattern that yields the best results?
4. How confident is CLIP when making correct and incorrect predictions?

For each one of the thirteen experiments, the same prompt variation was consistently applied on every class, except for the last experiment, where we prompted CLIP with the best performing prompt on each class. We are interested in the progression of the F-score per class. The curve should follow a consistent pattern over all classes, which should also be consistent with the average accuracy curve.

### RESULTS

Our initial emphasis is on the simplest prompt, namely the *class names*. This prompt type yields an average accuracy of **61.93%**, with the lowest F-score being 0% for the class *title*, as illustrated in Figure 3.2a. In fact, as can be observed in Figure 3.2b, all titles are misclassified as logos.




(A) F-score per class.

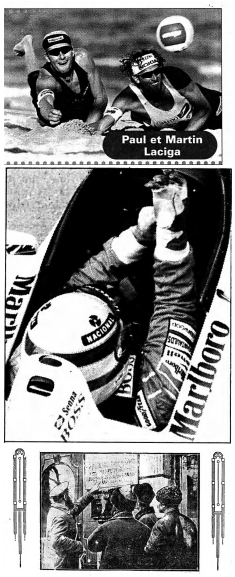
(B) Confusion matrix.

**FIGURE 3.2**
CLIP zero-shot classification results on *class name* prompt.

This result is not surprising given that the images for this class are mainly title-font single letters and not complete titles. We can safely affirm that this low score is due to the class name being inaccurate, more than the capability of the model. As presented in Figure 3.5, changing the prompt from *title* to *title-font letter* increases the F-score from 0% to 83%.

We notice in Figure 3.2b that the majority of misclassifications are false positive for the class *logo*. Mostly images labelled as *drawing*, *photo*, and *title* are misclassified as a *logo*. We can see a few examples of this type of misclassification in Figure 3.3.


(A) Drawings




(B) Photos

(C) Titles

**FIGURE 3.3**
Examples of false positives for *logo* using the *class name* prompt.

Moving forward to the next experiments, we obtain the holistic results for all prompts presented in Figure 3.4, which will enable us to answer the questions previously stated.

Looking at Figure 3.4a, we notice a general pattern in the F1 of the classes, revealing the best-performing prompts. This pattern is well reflected in Figure 3.4b, illustrating the average accuracy per prompt over all classes. Moreover, it overlaps with the pattern observed on the confidence level. This behaviour is favourable, as the model's confidence level is directly correlated to its accuracy.

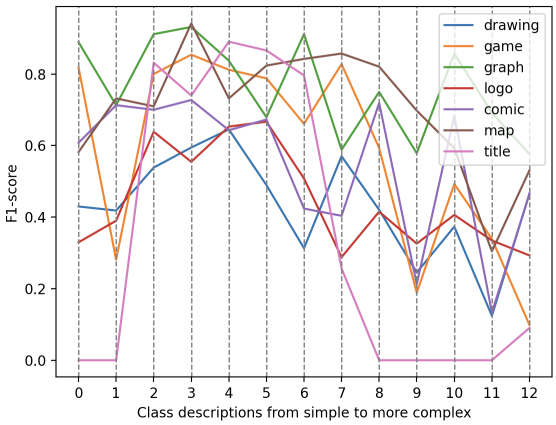
We notice that CLIP works well on the four specification prompts summarised in Table 3.2 along with their performance. The structure of these prompts is simpler than the paraphrasing prompts. The idea is to concisely express the essence of the class, exhaustively but with as less wording as possible. Moreover, CLIP is sensitive to changes in punctuation and connectors, as the *disjunction of noun phrases* prompt performs better than the *noun phrases separated by comas* prompt, even though the same noun phrases are used for both prompts.

Finally, CLIP is able to understand nuances of words as long as they are not too abstract. It can associate a word to a general concept if the word can be put out of its context and still hold the same meaning. As seen in Figure 3.4, it performs poorly on the metaphorical definitions of the classes, which correspond to prompt 12.

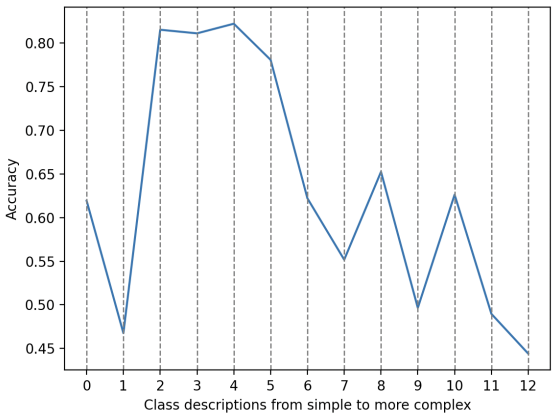
| **Prompt structure** | **Average accuracy** |
|---|---|
| noun phrase | 79.45% |
| adjective 1 + adjective 2 + noun phrase | 78.21% |
| disjunction of noun phrases | **82.21%** |
| noun phrases separated by comas | 78.07% |

**TABLE 3.2**
Average accuracy of the best performing prompt variations.

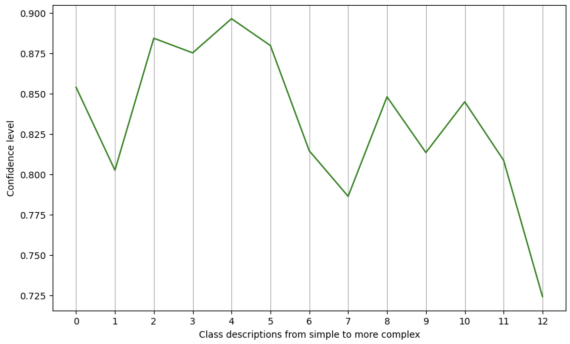
The best average accuracy is **82.21%** , when prompting CLIP with disjunctions of noun phrases.
Can we confidently assert that the *disjunction of noun phrases* prompt generally yields the best perform-




(A) F1 per class for each one of the 13 prompts.




(B) Average accuracy per prompt.




(C) Average confidence level per prompt.

**FIGURE 3.4**
CLIP zero-shot classification results on 13 prompt variations.

ance? To answer this question, we tried a new variation where we took the best performing prompt for each class. For every class, Figure 3.5 illustrates the difference in F-score between each type of variation and prompt 0, i.e. the class names. We notice that the highest increase always occurs when prompting CLIP with one of the four specification prompts, indexed from 2 to 5 in Figure 3.5.

The results show that the average accuracy stagnates in the range of 82%. Therefore, any of the four specification prompts is well understood by CLIP. Moreover, **CLIP performs significantly better when the specifications are exhaustive**.

We are left with the task of narrowing down the options and getting a more specific guideline to prompt CLIP optimally. The hypothesis that we will support in the following section is that the difference in performance between the specification prompts depends on factors outside the grammatical structure of the prompt.

We, thus, mainly focus on the influence of lexical choices. In order to do this, we study misclassifications and experiment on a selection of prompts by strategically altering the lexicon and analysing the results.

### 3.1.2 EXPERIMENTING WITH LEXICAL CHOICES

In this section, we have a closer look at the most recurrent misclassifications with the best results achieved yet, namely prompting CLIP with disjunction of noun phrases. We choose to focus specifically on the classes *comic* and *drawing* due to their interesting relationship. A *comic* is a subset of drawings, as it

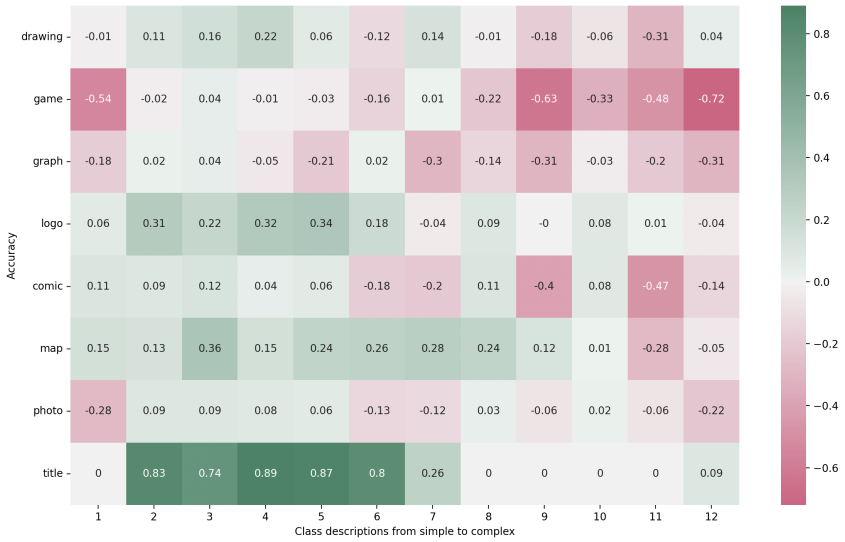
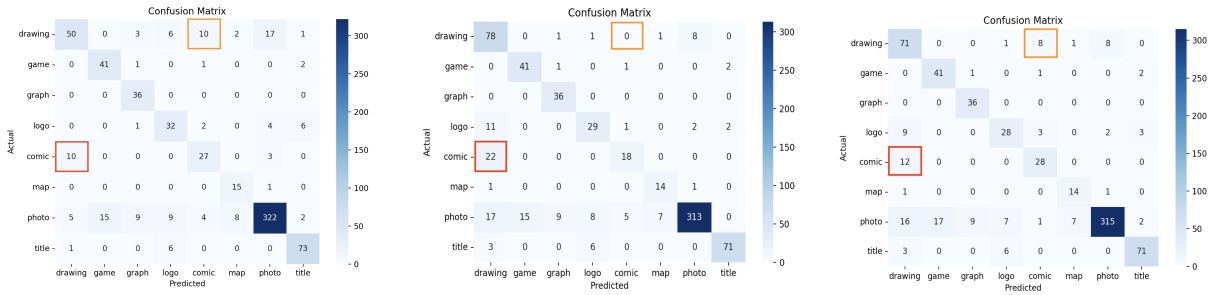



**FIGURE 3.5**
Differences in F-score between *class name* prompt and other prompt variations.

involves creating images through the art of drawing. However, not all drawings can be classified as comics, as comics typically involve a sequential narrative format with associated text or dialogue.

## RESULTS

We notice in Figure 3.6a that within the *comic* class, there are ten instances of false positives of comics, marked in orange, where drawings are misclassified as comics. On the other hand, there are ten instances of false negatives of comics, marked in red, where comics are misclassified as drawings.




**(A)** Confusion matrix with no lexicon alteration.

**(B)** Confusion matrix when specifying that a drawing does not contain text.

**(C)** Confusion matrix when specifying that a comic contains text.

**FIGURE 3.6**
Progression of the confusion matrix using the *disjunction of noun phrases* prompt when altering the lexicon of the prompts for *comic* and *drawing*.

To attempt fixing the false positives, we specify in the prompt for the class *drawing* that a drawing does not contain text. Thus, we modify the prompt from *"illustration or cartoon or caricature or descriptive drawing"* to *"non-textual illustration or cartoon or caricature or descriptive drawing"*.

The results in Figure 3.6b show that this specification decreases the misclassification of drawings as comics to 0 instances. A few images previously misclassified that are now classified correctly are shown in Figure 3.7.

**FIGURE 3.7**
Drawings misclassified as comics before specifying that a drawing does not contain text.

However, looking back at the misclassifications of comics as drawings, we see that they have increased from 10 to 22 instances. To attempt fixing this, we now specify in the prompt for the class **comic** that a comic does contain text. Thus, we modify the prompt from *"entertaining traditional comic"* to *"entertaining traditional comic including text"*. Doing this decreases the misclassifications studied from 22 to 12 instances, as seen in Figure 3.6c. A few images previously misclassified that are now classified correctly are shown in Figure 3.7.


**FIGURE 3.8**
Comics misclassified as drawings before specifying that a comic contains text *and* a drawing does not contain text.

Implementing the lexical modifications stated above yields a new accuracy of **83.33%**. From these results, we can conclude that **the prompts of the classes should be mutually exclusive**.

### 3.1.3 CLIP'S MULTILINGUAL ABILITIES

In this section, we explore CLIP's ability to generalise across languages, which is essential for its application in multilingual contexts and for inferring insights about the composition of its training data. To ensure a comprehensive evaluation, we selected a diverse range of languages from distinct families, as depicted in the leaves of the language tree shown in Figure 3.9. This approach allows us to investigate CLIP's performance across a variety of linguistic structures and vocabularies[2].

The languages chosen represent major families, each with its unique characteristics:

- Sino-Tibetan: Represented by Mandarin, this family is known for its tonal nature and logographic writing system, providing a test for CLIP's ability to handle non-alphabetic scripts and tonal distinctions.

[2]The translations of the prompts were made using the Python *translate* library `https://pypi.org/project/translate/`.

- Slavic: With Russian as the representative, we aim to test CLIP's handling of Cyrillic script and the complex inflectional morphology typical of Slavic languages.
- Germanic: English and German from this family allow us to evaluate CLIP's performance in widely spoken languages with rich resources, possibly reflecting the model's training data.
- Latin: French and Spanish represent Romance languages derived from Latin, each with its own evolution and linguistic nuances.
- Semitic: Arabic and Hebrew, with their right-to-left script and root-based morphology, offer a distinct challenge, testing the model's capacity to deal with non-Latin scripts and unique linguistic structures.

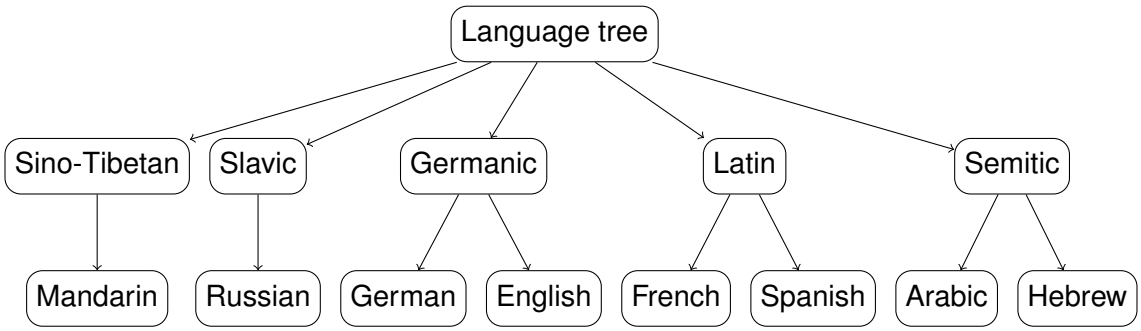



**FIGURE 3.9**
Language tree.

### RESULTS

Our analysis reveals that English is the most proficient language, with German, Spanish, and French also demonstrating notably higher performance compared to the other assessed languages. German, belonging to the same Germanic family as English, achieves a peak performance of **64.69%** using the *specification* prompt, which is structured as *"This" + class name + verb + preposition*.

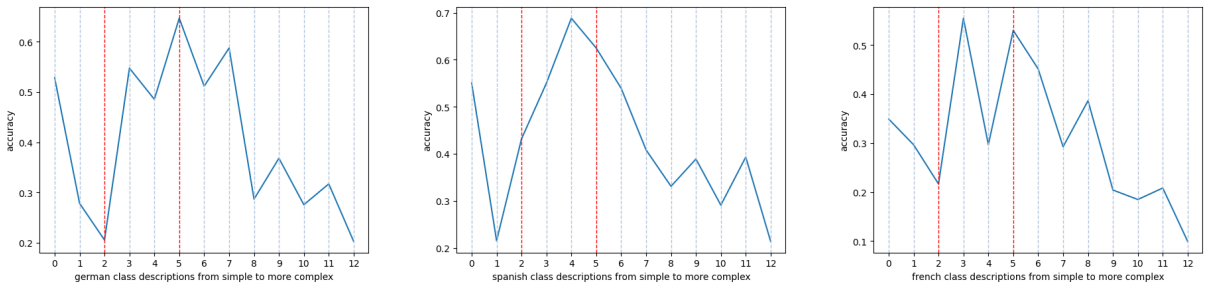



**FIGURE 3.10**
Average accuracies across the thirteen prompts translated in German, Spanish and French.

French and Spanish are both in the Latin family language. The peak performance is **68.83%** for Spanish, using the *disjunction of noun phrases* prompt, just like with the English prompts. As for French, it is **55.45%**, using the *noun phrases separated by comas* prompt.

It is noteworthy that the pattern observed in the previous section largely persists across these languages, with the specification prompt consistently emerging as the most effective across the three mentioned languages. This trend is further illustrated in Figure 3.10, which showcases the average accuracy progression across the various prompts when translated into German, Spanish, and French.

Conversely, languages from the Sino-Tibetan, Slavic, and Semitic families did not perform as strongly, with their peak average accuracy hovering around **35%**. Given this data, it is challenging to ascertain a clear pattern or identify the most effective prompts for these languages.

Based on these observations, it is reasonable to hypothesize that CLIP's training predominantly involved image and text pairs with text in Germanic and Latin languages. This insight could significantly influence future training and application strategies for multilingual vision-language models.

### 3.1.4 DISCUSSION & CONCLUSIONS

In our analysis of CLIP's performance, we deduce that the model excels when presented with exhaustive and mutually exclusive prompts for classification tasks, suggesting a potential competitive edge against state-of-the-art fine-tuned models. A detailed comparison will be presented in a subsequent section of this report.

We also observed that the confidence level associated with CLIP's predictions serves as a critical indicator of prompt effectiveness. Lower confidence levels are indicative of less accurate results, whereas higher confidence levels align with more accurate outcomes. This is a useful relationship for prompt engineering. Furthermore, our study highlights that CLIP exhibits optimal performance when prompted in English, which is likely reflective of the dominant language in its training data. While it demonstrates a reasonable understanding of Latin and Germanic languages, its proficiency diminishes significantly when confronted with other language families.

## 3.2 FLAMINGO

Flamingo is a model developed by Google DeepMind (Alayrac et al. 2022), which combines large language models with visual inputs designed to process both text and images simultaneously, allowing it to answer questions or perform tasks that rely on understanding both textual and visual information. The unique aspect of Flamingo is its ability to handle multimodal tasks that require understanding and integrating both text and visual data. This includes tasks like image captioning, visual question answering, and multimodal reasoning. Flamingo is built upon a foundation of large-scale Transformer-based language models. It extends these models' capabilities by including a mechanism for integrating visual information, essentially training the model to understand and process images in the context of related text.

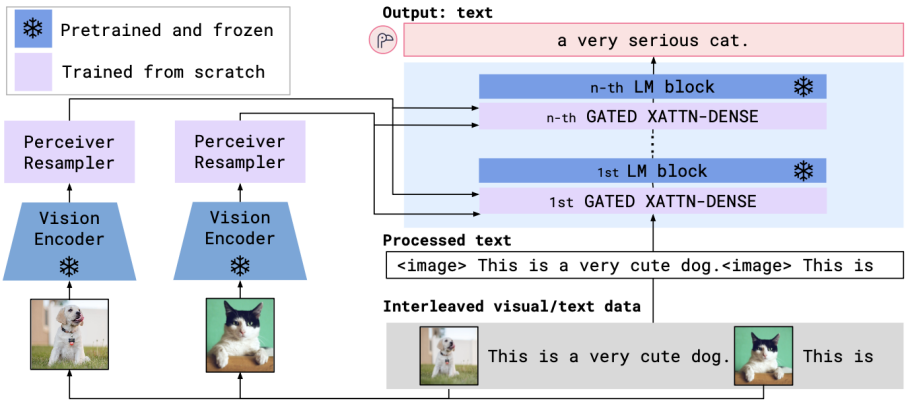



**FIGURE 3.11**
Architecture of Flamingo from Alayrac et al. 2022.

Its applications are broad and include performing tasks that require understanding the content of an image

in the context of a question or description, explaining visual scenes, and answering questions that rely on both visual and textual understanding.

As presented in Figure 3.11, Flamingo works by accepting interleaved inputs in the form of text and images. The text is parsed from the inputs and the images replaced by placeholders. The images are then extracted from the input, parsed through a frozen vision encoder, then mapped through the perceiver resampler to produce a fixed number of visual tokens per input. The tokens are used as inputs to cross attention layers to condition a language model that has taken as input the text sequence. The model finally proceeds with the completion of the partial text. It is worth noting that the vision encoder is pretrained as a dual encoder using contrastive loss employed by CLIP.

Flamingo is a particularly interesting model to study because of its flexible architecture. In fact, CLIP's architecture lacks flexibility, in the sense that the model can select the best image and text pairings, but does not have the ability to generate language. It is thus not suited for open-ended tasks like captioning or visual question answering. Flamingo, on the other hand, is able to perform open-ended tasks. We will particularly explore its captioning capabilities later in this report. But first, we explore its classification abilities.

**HYPERPARAMETERS** For the implementation of Flamingo, since the model is not open-sourced by DeepMind Google, we worked with the unofficial implementation by Awadalla 2022, (*CLIP repository* 2022) and took inspiration from the (Wang 2022) repository.

### 3.2.1 PROMPTING FLAMINGO FOR ZERO-SHOT & FEW-SHOT IMAGE CLASSIFICATION

It is possible to perform classification using Flamingo by inserting the class name into text templates of the form "a photo of a *class*", then selecting the text that is assigned the highest similarity to the image embedding by the model. This is done using maximum likelihood. However, since this would be equivalent to repeating the same experiment as previously done with CLIP, we focus on evaluating the generative capabilities of Flamingo.

To facilitate this, we employ four distinct types of prompts, each designed to elicit a specific kind of response from Flamingo. These prompts are the beginnings of sentences that the model will complete, providing predictions that can be post-processed to extract class information.

| Index | Prompt |
|---|---|
| 1 | "An image of" |
| 2 | "This image can be classified as" |
| 3 | "Keywords describing this image are" |
| 4 | "The type of this image is" |

**TABLE 3.3**
Prompt variations for classification using Flamingo.

The prompts are outlined in Table 3.3 and explained below:

- *"An image of"*: This is a simple and open-ended prompt, recommended by (Alayrac et al. 2022), encouraging the model to freely describe what it sees in the image. The generative nature of Flamingo can lead to diverse responses, which can then be analysed to infer the image's class.
- *"This image can be classified as"*: This prompt is more direct and asks the model to categorize the image explicitly. It is designed to assess how well Flamingo can perform explicit classification tasks in a generative manner.

- *"Keywords describing this image are"*: By asking for keywords, this prompt aims to extract specific descriptors from Flamingo that can be indicative of the image's class. This can be particularly useful for understanding the model's perception and the primary elements it focuses on when viewing an image.
- *"The type of this image is"*: Similar to the second prompt, this one also seeks a direct classification. However, the slight variation in wording allows us to compare the effects of phrasing on Flamingo's classification accuracy and consistency.

**FLAMINGO ANSWER POST-PROCESSING** The post-processing strategy employed consists in transforming the original captions into a single-word representation corresponding to the predicted class. Given a single caption, the algorithm iterates through each word and maps it to the appropriate class based on a predefined dictionary shown in Table 3.4.

| Class | Keyword list |
|---|---|
| drawing | drawing, illustration, fashion, ornamental, cartoon |
| game | puzzle, sudoku, chess, crosswords, poker, card games, grid, wordsearch |
| graph | chart, graph, plot, histogram, diagram, graphic |
| logo | logo, ad, advertisement, publicity, symbol, label, emblem |
| comic | comic, cartoon |
| map | map, blueprint, plan |
| photo | photograph, photo, view, close-up |
| title | title, letter, C, O, N, F, E, D, R, Confédéré, monogram |

**TABLE 3.4**
Dictionnary used for the post-processing of predictions.

This dictionary was created based on the class descriptions made when the dataset was annotated by Conti 2022, and by interactively adding synonyms when observing the model's responses. Subsequently, it handles all possible cases of duplicated or conflicting classes. There could be more than one predicted class if the caption contains keywords from different classes. In this case, if one of them is *drawing*, then the algorithm chooses the other class. This is because, as stated earlier, many classes are subsets of a *drawing*. For example, a comic is a drawing, but a drawing is not necessarily a comic.

Otherwise, the algorithm chooses the class that has the maximum number of occurrences, if there is such a class. If not, the algorithm dynamically prompts Flamingo to settle between the classes, ensuring a more accurate and refined result. The prompt used to do so is the following: "*Between the classes __ and __, this is a*".

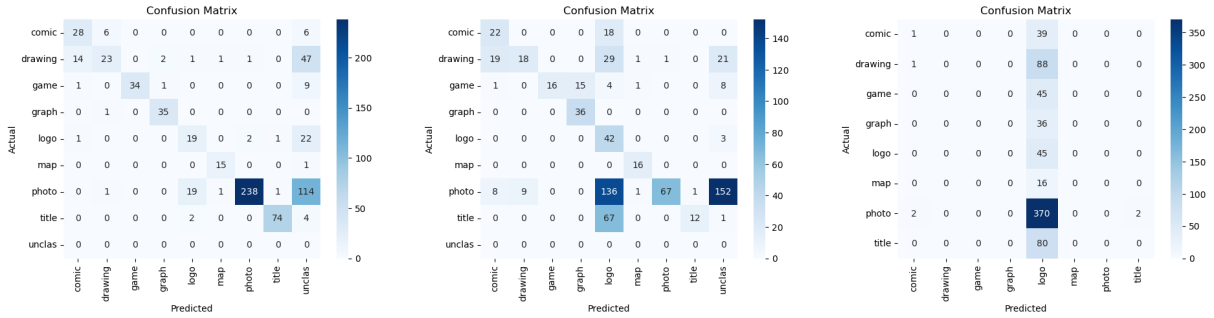
It recursively processes the new generated prompts, following this same algorithm. This strategy is applied to all generated captions, and aims to enhance the precision of the model's classification output by intelligently handling ambiguities and optimizing the query prompts for improved results.

### RESULTS

Prompting Flamingo with a 0-shot prompt highlights its proficiency in describing an image concisely, and with precise words. This is being reflected in a very high precision for all classes. However, we observe a very low recall since the generated texts are so distinctive that it is hard to include all possible keywords in a dictionary. For this reason, there are a lot of misclassifications. This particularly happens for the class *photo* since the model describes what it sees on the photograph instead of recognizing that it is a photograph. Prompt 2, namely *"This image can be classified as"* mitigates this behaviour since it explicitly asks to *classify* the image, to which the model sometimes responds *photograph*. Concerning the

0-shot experiment on prompt 2, there is a total of 202 out of 725 unclassified images, 114 of which are photographs, and only 53 misclassifications. The repartition of these misclassifications can be seen in Figure 3.12a.

This behaviour immediately changes when inputting few-shot prompts. As illustrated in Figures 3.12b and3.12c, the misclassifications increase significantly, as the number of unclassified images decreases from 202 to 185 in 1-shot, then to 0 in 2-shots. On the other hand, misclassifications increase from 53 to 332 to 680 in 2-shots, where all images are classified as logos. Looking at all few-shot results across prompts, it seems that Flamingo picks up on the pattern of the prompt but fails to complete it correctly.




(A) 0-shot (B) 1-shot (C) 2-shot

**FIGURE 3.12**
Confusion matrices across number of shots prompting Flamingo with *"This image can be classified as a"*.

We choose to look at the macro-average to assess the results, presented in Table 3.5, since it provides a balanced view of performance across all classes.

| **Prompt** | **Macro-average** | | |
|---|---|---|---|
| | **0-shot** | **1-shot** | **2-shot** |
| 1 | 48.97% | 41.49% | 25.84% |
| 2 | 65.82% | 43.04% | 2.04% |
| 3 | 43.45% | 55.79% | 38.52% |
| 4 | 42.94% | 56.03% | 20.26% |

**TABLE 3.5**
Macro-average per prompt using Flamingo.

### 3.2.2 DISCUSSION & CONCLUSIONS

To gain more insights on the capabilities of Flamingo and its behaviour, we perform a quantitative and a qualitative analysis of the texts it generated. The predicted texts are displayed on the table accessible at `https://github.com/dh-epfl-students/dhlab-image-captioning/blob/main/FLAMINGO/class_summary.html`).

#### QUANTITATIVE ANALYSIS

The quantitative analysis performed here focuses on the frequency of the texts predicted by Flamingo. Since all images are different, it is likely that the captions are distinct even if two images belong to the same class. The desired behaviour is that the number of different captions decreases as we increase the number of shots. We observe in Figure 3.13 that this is indeed what happens, however, the performance decreases as seen in the previous section.

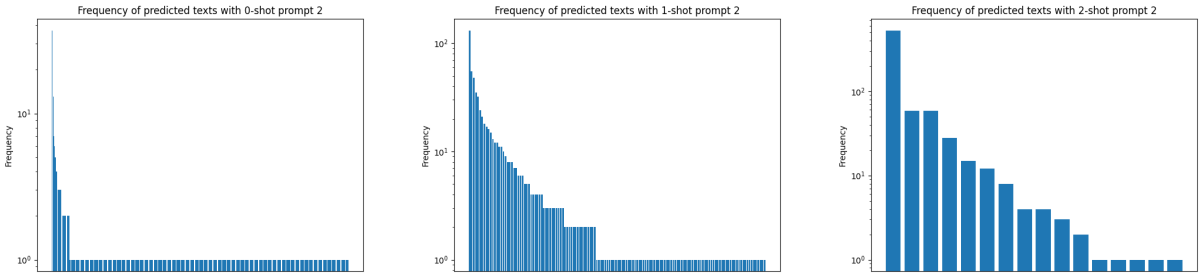



**FIGURE 3.13**
Frequency of prompts on 0-shot, 1-shot and 2-shot using the prompt *"This image can be classified as a"*.

We also notice that, as we increase the number of shots, the generated texts become more repetitive. The top-3 repeated texts across number of shots on prompt 2 are shown in Table 3.6.

| 0-shot | 1-shot | 2-shot |
|---|---|---|
| crossword puzzle | This image can be classified as a logo. | This image can be classified as a logo. This image can be classified as a game. This image can be classified |
| comic strip | This image can be classified as a logo. This image can be classified as a painting. This image can be classified | This image can be classified as a comic. This image can be classified as a logo. This image can be classified |
| bar graph | This image can be classified as a photograph | This image can be classified as a graph. This image can be classified as a logo. This image can be classified |

**TABLE 3.6**
Top 3 predictions across the number of shots for the prompt *"This image can be classified as a"*.

#### QUALITATIVE ANALYSIS

When looking at the generated texts, we notice that for the same image, the model generates a sequence of different predictions following the prompt pattern, for example: “This image can be classified as a logo. This image can be classified as a game. This image can be classified". This is an undesired behaviour and can hypothetically be explained by the fact that few-shot prompting confuses the model by making the prompt very long.

We can confidently assert that the model detects well the elements of an image, and can differentiate between objects, people and animals. However, it lacks the ability to generate mutually exclusive predictions, for example: “This image can be classified as a graph. This image can be classified as a logo”. Moreover, it could benefit from a better understanding of the prompt and the context. It should recognize that the sentence completion is expected to provide a single, coherent suggestion rather than a list of possibilities.

## 3.3 COMPARING CLIP, FLAMINGO, AND VGG-16

VGG-16 is a convolutional neural network architecture that was a notable development in the field of deep learning and computer vision, and it achieved high levels of accuracy in the ImageNet competition, which is a benchmark in image classification and recognition.

We compare the three models’ best performance against each other. For VGG-16, the result presented

in Table 3.7 was obtained by training the model on the dataset using an SGD optimizer, with batch size 32 and learning rate 0.01 (Conti 2022). On the other hand, CLIP's result was obtained by prompting the model with disjunction of noun phrases on a closed-ended task. Finally, the result using Flamingo was achieved by prompting Flamingo on an open-ended task, inputting *"This image can be classified as a"* and post-processing the predictions to map them to the desired classes. It is important to mention that the results obtained using VGG-16 were achieved on the complete dataset, i.e. including the classes *diverse* and *other*, which were omitted in the context of our experiments on CLIP and Flamingo.

| | **VGG-16** | **CLIP** | **Flamingo** |
|---|---|---|---|
| **Average accuracy** | **89.14%** | 83.33% | 64.28% |

**TABLE 3.7**
Comparison table of average accuracies.

## 3.4 CONCLUSIONS

Upon examining the generated texts, a notable observation is the model's tendency to produce diverse predictions for the same image when prompted in a few-shot manner. This behaviour, seemingly triggered by the length of the prompt, is undesirable and requires further investigation.

Moving forward, our comparison of CLIP, Flamingo, and VGG-16 sheds light on their respective strengths and weaknesses. For CLIP to compete with fine-tuned models, it should be prompted optimally, following the patterns we have recognised to be effective through our experiments. This requires some prompt engineering, and opens up the possibility of an automatised way of improving the prompts. On the other hand, since classification results were derived from prompting Flamingo on an open-ended task, they underscore the true classification capability of the model, but offer us more understanding of its strengths and limitations.

# 4 Image Captioning

In this chapter, we delve into the experimentation with Flamingo for zero-shot image captioning. We designed our study to evaluate Flamingo's capacity to generate accurate and contextually rich captions without prior training on specific image datasets. This approach is particularly challenging and promising, as it relies solely on the model's pre-existing knowledge and its ability to generalize from it.

## 4.1 Prompting Flamingo for zero-shot image captioning

To conduct a comprehensive evaluation, we employed three different prompts, as outlined in Table 4.1.

| Index | Prompt |
|---|---|
| 0 | "An image of" |
| 1 | "A complete caption for this image is:" |
| 2 | "Question: What can you say about the location and time of this image? Answer:" |

**Table 4.1**
Prompt variations for captioning using Flamingo.

Each prompt was carefully crafted to analyse various aspects of Flamingo's captioning abilities:

- *An image of:* This prompt is straightforward and open-ended, designed to assess the model's basic descriptive capabilities. It serves as a baseline to understand how Flamingo interprets and narrates the visual content without specific guidance.
- *A complete caption for this image is:*: With this prompt, we aim to direct the model to produce a more detailed and structured response. It's an attempt to encourage Flamingo to not only describe what's visible, but also to infer and articulate a more comprehensive understanding of the image.
- *Question: What can you say about the location and time of this image? Answer:*: This prompt is particularly aimed at evaluating Flamingo's ability to contextualize the image. By asking for the location and time, we are testing the model's capacity to deduce historical and geographical information, which often requires a higher level of inference and understanding.

We aim to evaluate the predicted captions based on distinctiveness, coherence, fluency, relevance, and completeness. Additionally, we are interested in the model's ability to contextualize an image in terms of time and place.

## 4.2 Discussion & Conclusions

We perform a qualitative analysis of the results by sampling five images per class and evaluating Flamingo's captioning abilities based on various aspects, inspired by the Fu et al. 2023 evaluation benchmarks. The

generated captions for all prompts are displayed on the table accessible at `https://github.com/dh-epfl-students/dhlab-image-captioning/blob/main/FLAMINGO/cap_summary.html`.

### QUALITATIVE ANALYSIS

The predictions are generally distinctive. Moreover, it generally recognizes the regions of the world on a map, in a picture with hints such as country flags or signs that require OCR. The generated text for a few maps illustrate this point and can be observed in Table 4.2.

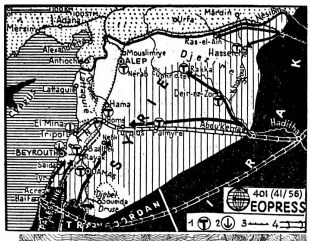
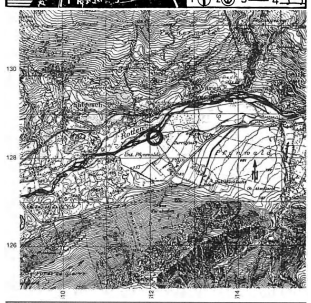
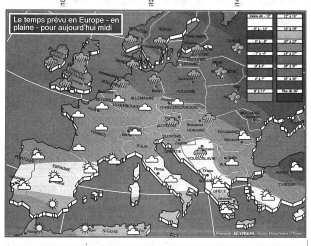
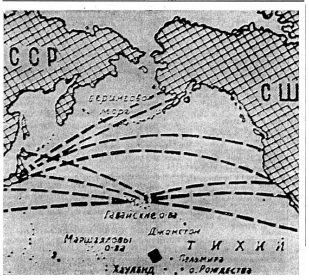
| Image | Caption | Analysis |
|---|---|---|
|  | “a map of the Middle East during the First World War.” | This is indeed a map of the Middle East. However, it is after WWI since Transjordan which appears at the bottom was only founded in 1921. |
|  | “the Ordnance Survey One-inch to the mile map of the Lake District, c.18” | The caption is false since this is a map of the surroundings of Salgesch in Switzerland. |
|  | "a map of the European continent showing the weather conditions for the month of May." | The caption appears accurate, inferring May from weather conditions, despite the absence of an explicit OCR indication. |
|  | "a map of the Soviet Union and its allies in the Second World War." | The map indeed includes a part of URSS, Canada and the USA, which were all allies in WWII. However, it also includes Japan, and the caption seems to miss what is happening in Hawaii. |

**TABLE 4.2**
Examples of generated captions for the prompt *"An image of"* on maps.

As for the timing, it is sometimes capable of accurately situating the image in the correct time period, but at other times, it makes false assumptions. As seen in Table 4.3, it recognizes that the dress is indeed from the 1920s, and that the *Special mode* logo is from 1970.

| Image | Caption | Image | Caption |
|---|---|---|---|
|  | Question: What can you say about the location and time of this image? Answer: It was taken in the 1920s. |   | Question: What can you say about the location and time of this image? Answer: It was taken in Paris, France, in the early 1970s. |

**TABLE 4.3**
Examples of correct temporal contextualization

However, the model is frequently incoherent. In fact, for the same image but different captions, it can generate contradictory information. This is illustrated using the captions of a map of Salgesch, Switzerland, in Table 4.4.

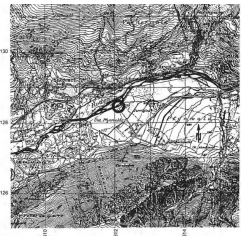
| Image | Caption 1 | Caption 2 | Caption 3 |
|---|---|---|---|
|  | A complete caption for this image is: Aerial view of the summit of Mount | the Ordnance Survey One-inch to the mile map of the Lake District, c.18 | Question: What can you say about the location and time of this image? Answer: This is a map of the location of the Battle of the Little Bighorn in 1876. |

**TABLE 4.4**
Example of an incoherency.

Furthermore, for a few images that take place in a general context, Flamingo tends to generate very specific captions. Our hypothesis from the sample observed is that if it recognizes a few elements that coincide with an image it was trained on, the model associates it to the paired caption it was trained on. Examples are shown in Table 4.5. It seems that the model lacks the ability to analyse correctly the context in a picture. Rather, it tries to find similarities between the training data and the given picture, then outputs the caption it was trained on.

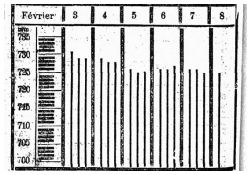
| Image | Caption | Image | Caption |
|---|---|---|---|
|   | An image of: Table 1 from the report of the Commission of Enquiry into the Condition of the Indian Population in [...] |  | A complete caption for this image is: The old school house, built in 1856, is now the home of the Historical Society. |

**TABLE 4.5**
Example of Flamingo assuming a context based on its training data.

# 5 Conclusions & Future Work

This project aimed at exploring models with the potential of automatically classifying and captioning historical images. These tasks are crucial for the digital preservation and accessibility of historical newspapers and efficiency of research workflows.

We observed that CLIP excels in image classification when prompted correctly, and we put forward a few guidelines to do so. Evaluating whether prompt engineering, prompt tuning or reinforcement learning (Deng et al. 2022) enable to systematically prompt CLIP optimally for maximized zero-shot results, and whether this offers significant resource saving compared to traditional fine-tuned machine learning models is an interesting research question. Whereas CLIP achieved state-of-the-art multimodal zero-shot performance at the time it was released in 2021 by OpenAI, it has now been outperformed by models such as BLIP-2 by Salesforce, GPT-4 by OpenAI, PaLM 2, LaMDA or more recently Gemini by Google. However, CLIP is still widely used and there has been research on how to use more recent models like GPT-4 to adapt CLIP to downstream tasks (Maniparambil et al. 2022), and on how to improve CLIP's performance by training it on diverse variants of each caption (Fan et al. 2023).

The primary goal of Flamingo was originally to generate accurate and informative descriptions for trending YouTube Shorts, which were then stored as metadata to enhance video categorization and searchability [1] . This systematic way of generating metadata could improve retrieval of specific information, enhancing the accessibility of these resources to researchers, historians, and the general public. Exploring whether this can work with historical videos, and how well it would work on noisy data, is an interesting open question.

[1] https://www.deepmind.com/blog/working-together-with-youtube?utm_source=linkedin&utm_medium=social&utm_campaign=YouTubeShorts

# Bibliography


Bisk, Yonatan et al. (2020). 'Experience Grounds Language'. In: *CoRR* abs/2004.10151. URL: https://arxiv.org/abs/2004.10151.

Radford, Alec et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision*. URL: https://arxiv.org/abs/2103.00020.

Alayrac, Jean-Baptiste et al. (2022). 'Flamingo: A Visual Language Model for Few-Shot Learning'. In: *Advances in Neural Information Processing Systems* 35, pp. 23716–23736.

Ramesh, Aditya et al. (2022). 'Hierarchical text-conditional image generation with clip latents'. In: *arXiv preprint arXiv:2204.06125* 1.2, p. 3.

*CLIP repository* (2022). OpenAI. URL: https://github.com/openai/CLIP.

*Zero-shot Image Classification with OpenAI's CLIP* (2022). Pinecone. URL: https://www.pinecone.io/learn/series/image-search/zero-shot-image-classification-clip/.

*Multi-modal ML with OpenAI's CLIP* (2022). Pinecone. URL: https://www.pinecone.io/learn/series/image-search/clip/.

Dosovitskiy, Alexey et al. (2020). 'An image is worth 16x16 words: Transformers for image recognition at scale'. In: *arXiv preprint arXiv:2010.11929*.

Awadalla, Anas (2022). *Open Flamingo*. https://github.com/mlfoundations/open_flamingo.

Wang, Phil (2022). *Flamingo - Pytorch*. https://github.com/lucidrains/flamingo-pytorch.

Conti, Pauline Isabela (2022). 'Historical Newspaper Image Classification'. In: URL: https://github.com/impresso/impresso-image-classification/tree/pauline.

Fu, Chaoyou et al. (2023). 'Mme: A comprehensive evaluation benchmark for multimodal large language models'. In: *arXiv preprint arXiv:2306.13394*.

Deng, Mingkai et al. (2022). 'Rlprompt: Optimizing discrete text prompts with reinforcement learning'. In: *arXiv preprint arXiv:2205.12548*.

Maniparambil, Mayug et al. (2022). 'Enhancing CLIP with GPT-4: Harnessing Visual Descriptions as Prompts'. In: *Advances in Neural Information Processing Systems* 35, pp. 23716–23736. URL: https://arxiv.org/abs/2307.11661.

Fan, Lijie et al. (2023). 'Improving CLIP Training with Language Rewrites'. In: *arXiv preprint arXiv:2305.20088*.