Phase 4 subtask 2 – Auto Caption Generation for Images

Concept understanding

The capacity to automatically generate meaningful captions for visual content without the need human writing is known as auto captioning. An AI algorithm creates a caption that is legible by humans by interpreting the visual elements of an image and translating them into natural language. Relying on only being watched, auto captioning makes it possible to comprehend visual information.

This is very useful for technical documents. Technical documents typically contain a large number of images, such as schematics, diagrams, layouts and the majority of these images have little to no extra descriptive text. As a result, users may find it challenging to decipher the images significance, by adding descriptions through auto captioning, users will be better able to comprehend and interpret the visual information, which will improve their ability to use the technical documentation.

Why Auto Captioning is so important in Multimodal Rag?

Most users in a multimodal Retrieve And Generate (RAG) System, start their queries with text-based requests. Unfortunately, this causes a problem when there are no text descriptions linked to the photographs, the end users result will not match due to this disparity between the text-based request and the visual nature of the asset.

These constraints are addressed in several ways via auto captioning. The user can now utilise a standard search technique "a text-based request" to locate a visual asset by turning it into a written description. Furthermore, by incorporating written descriptions that enable precise comprehension of how the visual asset relates to the request, the automatically generated captions offer more context than the picture embeddings.

Because of this, the system is able to accurately identify the right image in response to user queries such as "find a picture of the wiring for my electric motor" and return that image. As a result, when huge language models with visual context react to a request, users obtain more precise and understandable answers.

In conclusion, rather than treating a picture as a stand-alone entity, auto captioning makes it possible to employ images inside the full "knowledge element" of retrieval and reasoning.

How auto caption generation works?

In general, there is a logical development involved in the auto-generating caption process. The auto-generated captioning model first analyses the photos to find pertinent areas of interest. It can assess the photos using a variety of modalities, such as item detection, form analysis, spatial relationships between objects, etc., thanks to its deep learning capabilities.

An automatically generated caption explaining the image's contents is created when these features are assessed. The automatically generated captions will then be linked to the file location of the original image and handled as metadata. Later on, auto-generated captions can also be indexed and transformed into embeddings, which greatly simplifies their retrieval using a regular text-based search.

Model Used : BLIP

An auto-captioning generation model called BLIP (Bootstrapped Language–Image Pretraining) uses a Vision–Language Model to produce text from images by combining both languages and images into a single model. It is employed for a number of tasks, including creating captions for pictures and comprehending the relationship between images and words (i.e., how to associate a visual object with a description).

When utilising diagrams, blueprints, and other lined documents, BLIP excels at producing precise captions and comprehending the connections between text and images. As a result, BLIP is especially useful for producing user manuals, where blueprints and diagrams—rather than conventional photographs or images referred to as "real-world" images—provide much of the information required to explain how goods operate.

To sum up, BLIP employs an Image and Language Model to enhance caption generation, which will eventually allow for increased user engagement when reading manuals. The owner's manual from the manufacturer for that specific vehicle is in agreement with all of the information shown here.

BLIP is an all-inclusive auto-caption generator powered by the advancement of language and image models.

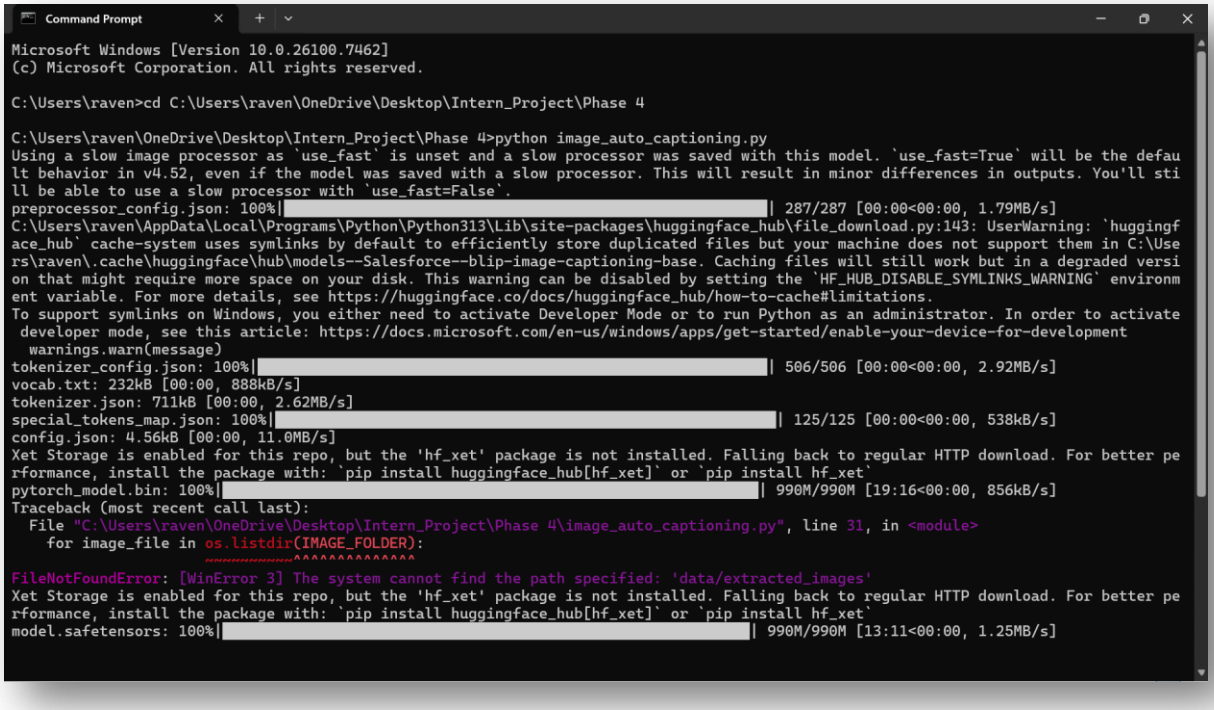
Limitations and Considerations

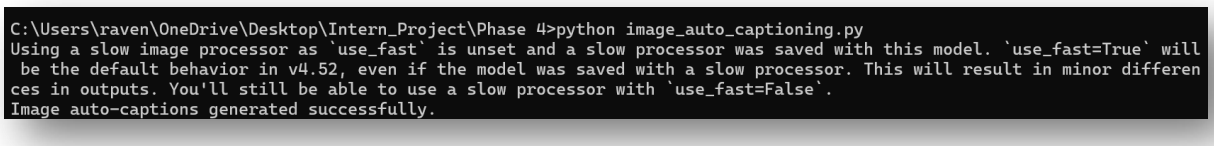
Auto captions are helpful, but they have certain drawbacks. For instance, when they generate captions for really sophisticated diagrams, they could be relatively "generic" and not accurately depict every potential component or specialised notion. This also applies to small text or fine-grained labelling in photos that the model might overlook or just partially read correctly. Optical Character Recognition (OCR) may therefore be required for diagrams containing a significant amount of written text in order to precisely transfer this information back into the original location within the image.

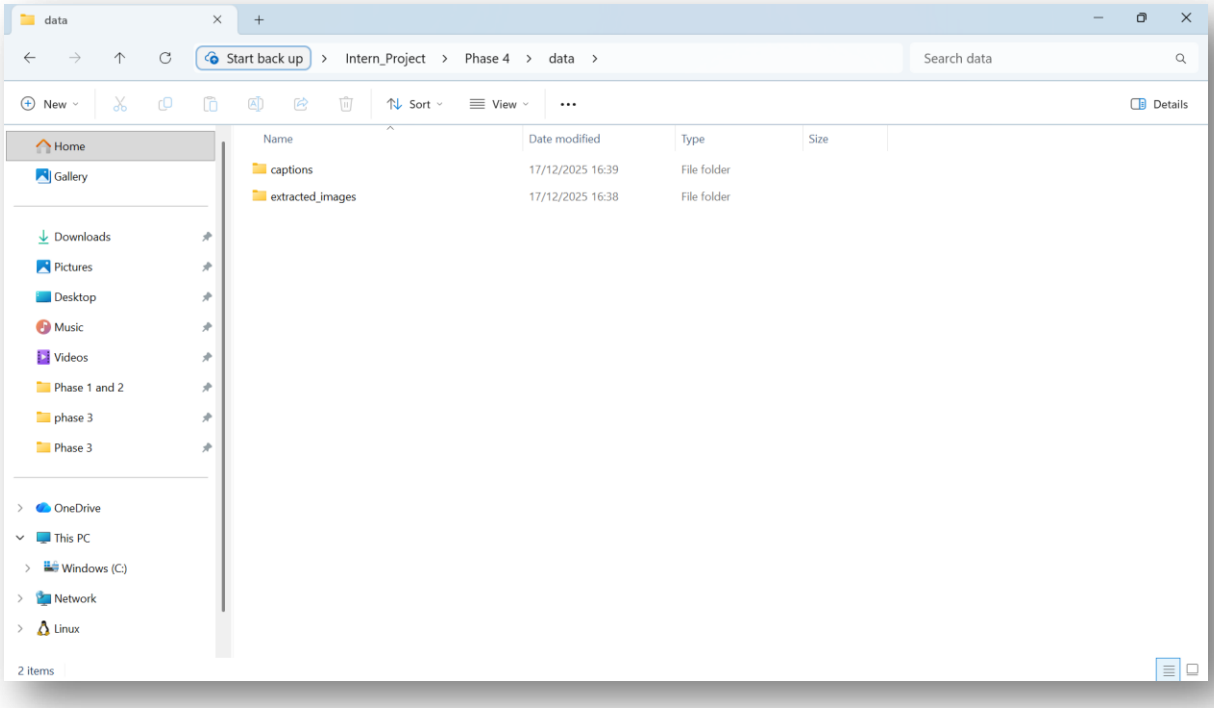
Because of this, the labels produced by auto captioning ought to be regarded as supplementary metadata rather than as conclusive facts.

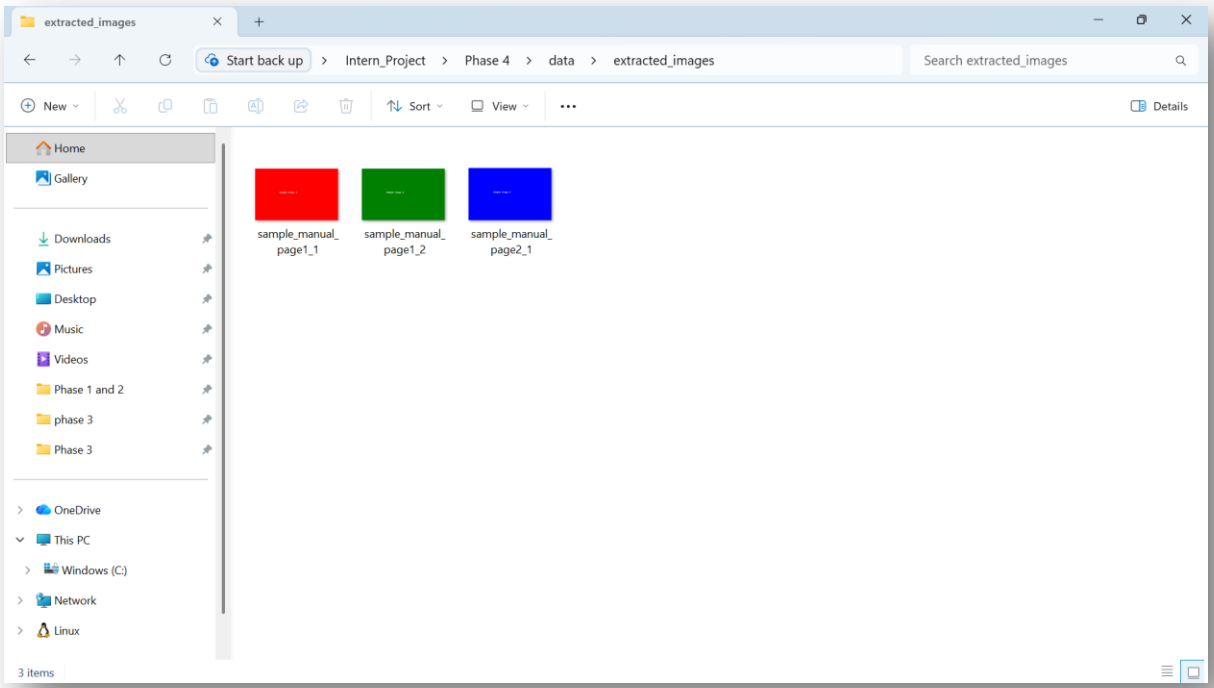
By making images easier to find, embed, and use within a digital reference system, auto-captioning continues to enhance the overall usability of images in multimodal search and retrieval despite these drawbacks.

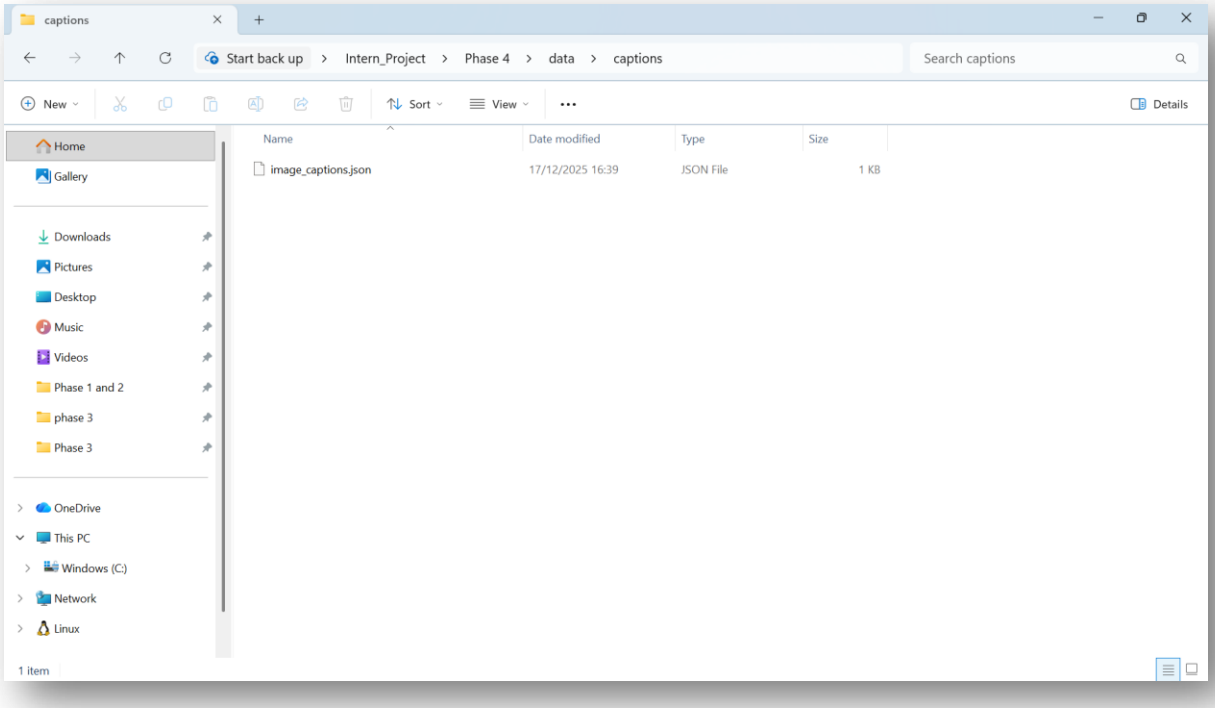
Testing


Command Prompt
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.
C:\Users\raven>cd C:\Users\raven\OneDrive\Desktop\Intern_Project\Phase 4
C:\Users\raven\OneDrive\Desktop\Intern_Project\Phase 4>python image_auto_captioning.py
Using a slow image processor as `use_fast` is unset and a slow processor was saved with this model. `use_fast=True` will be the defau
lt behavior in v4.52, even if the model was saved with a slow processor. This will result in minor differences in outputs. You'll sti
ll be able to use a slow processor with `use_fast=False`.
preprocessor_config.json: 100%| 287/287 [00:00<00:00, 1.79MB/s]
C:\Users\raven\AppData\Local\Programs\Python\Python313\Lib\site-packages\huggingface_hub\file_download.py:143: UserWarning: `huggingf
ace_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Use
rs\raven\.cache\huggingface\hub\models--Salesforce--blip-image-captioning-base. Caching files will still work but in a degraded versi
on that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environm
ent variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate
developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development
warnings.warn(message)
tokenizer_config.json: 100%| 506/506 [00:00<00:00, 2.92MB/s]
vocab.txt: 232kB [00:00, 888kB/s]
tokenizer.json: 711kB [00:00, 2.62MB/s]
special_tokens_map.json: 100%| 125/125 [00:00<00:00, 538kB/s]
config.json: 4.56kB [00:00, 11.0MB/s]
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better pe
rformance, install the package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
pytorch_model.bin: 100%| 990M/990M [19:16<00:00, 856kB/s]
Traceback (most recent call last):
File "C:\Users\raven\OneDrive\Desktop\Intern_Project\Phase 4\image_auto_captioning.py", line 31, in <module>
for image_file in os.listdir(IMAGE_FOLDER):
FileNotFoundError: [WinError 3] The system cannot find the path specified: 'data/extracted_images'
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better pe
rformance, install the package with: `pip install huggingface_hub[hf_xet]` or `pip install hf_xet`
model.safetensors: 100%| 990M/990M [13:11<00:00, 1.25MB/s]


C:\Users\raven\OneDrive\Desktop\Intern_Project\Phase 4>python image_auto_captioning.py
Using a slow image processor as `use_fast` is unset and a slow processor was saved with this model. `use_fast=True` will
be the default behavior in v4.52, even if the model was saved with a slow processor. This will result in minor differen
ces in outputs. You'll still be able to use a slow processor with `use_fast=False`.
Image auto-captions generated successfully.


data
Start back up
Intern_Project
Phase 4
data
Search data
New
Sort
View
Details
Home
Gallery
Downloads
Pictures
Desktop
Music
Videos
Phase 1 and 2
phase 3
Phase 3
OneDrive
This PC
Windows (C:)
Network
Linux
Name
Date modified
Type
Size
captions
17/12/2025 16:39
File folder
extracted_images
17/12/2025 16:38
File folder
2 items


extracted_images
Start back up
Intern_Project
Phase 4
data
extracted_images
Search extracted_images
New
Sort
View
Details
Home
Gallery
Downloads
Pictures
Desktop
Music
Videos
Phase 1 and 2
phase 3
Phase 3
OneDrive
This PC
Windows (C:)
Network
Linux
sample_manual_page1_1
sample_manual_page1_2
sample_manual_page2_1
3 items


captions
Start back up
Intern_Project
Phase 4
data
captions
Search captions
New
Sort
View
Details
Home
Gallery
Downloads
Pictures
Desktop
Music
Videos
Phase 1 and 2
phase 3
Phase 3
OneDrive
This PC
Windows (C:)
Network
Linux
Name
Date modified
Type
Size
image_captions.json
17/12/2025 16:39
JSON File
1 KB
1 item