Qwen / **Qwen2.5-VL-7B-Instruct** like 1.27k

Follow Qwen 51.3k

Image-Text-to-Text | Transformers | Safetensors | English | qwen2_5_vl

image-to-text | multimodal | conversational | text-generation-inference

arxiv:2309.00071 | arxiv:2409.12191 | arxiv:2308.12966 | License: apache-2.0

Train | Deploy | Use this model

Model card | Files xet | Community 59

Downloads last month

4,324,866

Safetensors

Model size 8.29B params | Tensor type BF16 | Chat template | Files info

Inference Providers NEW

Image-Text-to-Text

Examples

Input a message to start chatting with **Qwen/Qwen2.5-VL-7B-Instruct**.

Your sentence here...

View Code

Run 15,000+ Models Instantly

Inference Providers let you run inference on thousands of models served by our partners using a simple, unified, OpenAI-compatible serverless API (Learn more).

`Qwen/Qwen2.5-VL-7B-Instruct` is supported by the following Inference Providers:

OVHcloud AI Endpoints | Hyperbolic

View API Code | Dismiss

Model tree for `Qwen/Qwen2.5-VL-7B-Instruct`

- Adapters — 99 models
- Finetunes — 714 models
- Merges — 1 model
- Quantizations — 101 models

Spaces using `Qwen/Qwen2.5-VL-7B-Instruct` 100

Collection including `Qwen/Qwen2.5-VL-7B-Instruct`

Qwen2.5-VL Collection

Vision-language model series based on Q… • 11 items • Updated Jul 21 • △ 538

Qwen2.5-VL-7B-Instruct

💜 Qwen Chat

Introduction

In the past five months since Qwen2-VL's release, numerous developers have built new models on the Qwen2-VL vision-language models, providing us with valuable feedback. During this period, we focused on building more useful vision-language models. Today, we are excited to introduce the latest addition to the Qwen family: Qwen2.5-VL.

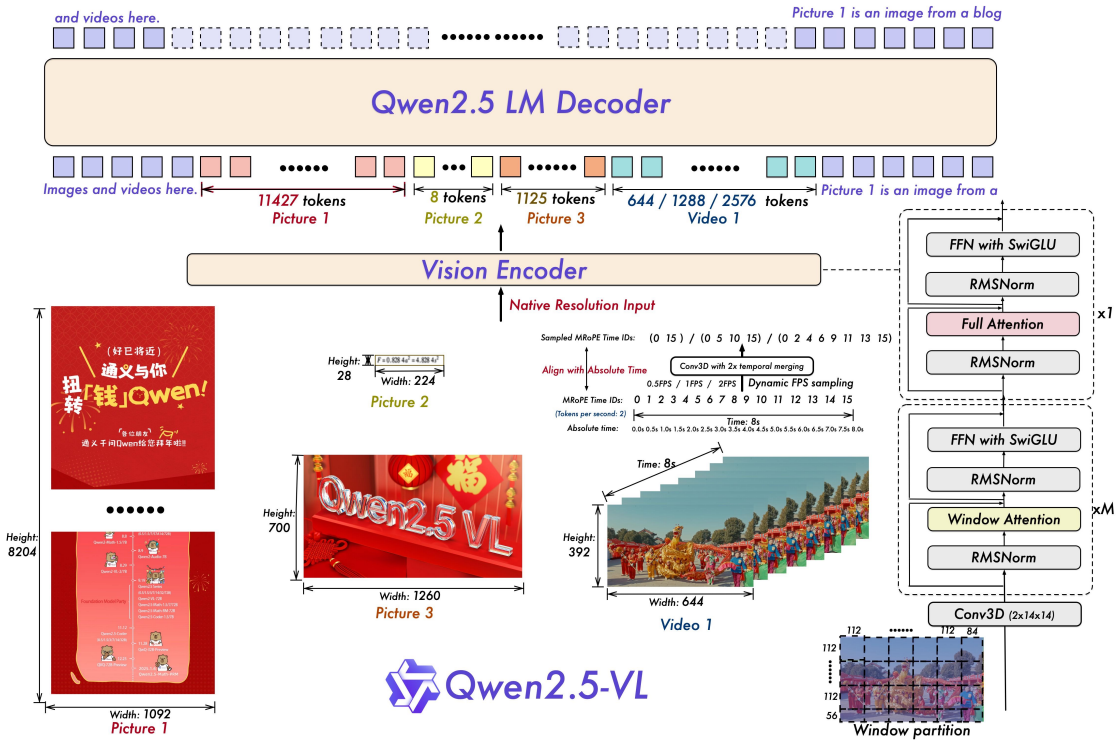
Key Enhancements:

- **Understand things visually**: Qwen2.5-VL is not only proficient in recognizing common objects such as flowers, birds, fish, and insects, but it is highly capable of analyzing texts, charts, icons, graphics, and layouts within images.
- **Being agentic**: Qwen2.5-VL directly plays as a visual agent that can reason and dynamically direct tools, which is capable of computer use and phone use.
- **Understanding long videos and capturing events**: Qwen2.5-VL can comprehend videos of over 1 hour, and this time it has a new ability of cpaturing event by pinpointing the relevant video segments.
- **Capable of visual localization in different formats**: Qwen2.5-VL can accurately localize objects in an image by generating bounding boxes or points, and it can provide stable JSON outputs for coordinates and attributes.
- **Generating structured outputs**: for data like scans of invoices, forms, tables, etc. Qwen2.5-VL supports structured outputs of their contents, benefiting usages in finance, commerce, etc.

Model Architecture Updates:

- **Dynamic Resolution and Frame Rate Training for Video Understanding**:

We extend dynamic resolution to the temporal dimension by adopting dynamic FPS sampling, enabling the model to comprehend videos at various sampling rates. Accordingly, we update mRoPE in the time dimension with IDs and absolute time alignment, enabling the model to learn temporal sequence and speed, and ultimately acquire the ability to pinpoint specific moments.



- **Streamlined and Efficient Vision Encoder**

We enhance both training and inference speeds by strategically implementing window attention into the ViT. The ViT architecture is further optimized with SwiGLU and RMSNorm, aligning it with the structure of the Qwen2.5 LLM.

We have three models with 3, 7 and 72 billion parameters. This repo contains the instruction-tuned 7B Qwen2.5-VL model. For more information, visit our Blog and GitHub.

Evaluation

Image benchmark

Benchmark	InternVL2.5-8B	MiniCPM-o 2.6	GPT-4o-mini	Qwen2-VL-7B	Qwen2.5-VL-7B
$MMMU_{val}$	56	50.4	**60**	54.1	58.6
$MMMU\text{-}Pro_{val}$	34.3	-	37.6	30.5	41.0
$DocVQA_{test}$	93	93	-	94.5	**95.7**
$InfoVQA_{test}$	77.6	-	-	76.5	**82.6**
$ChartQA_{test}$	84.8	-	-	83.0	**87.3**
$TextVQA_{val}$	79.1	80.1	-	84.3	**84.9**
OCRBench	822	852	785	845	**864**

Benchmark	InternVL2.5-8B	MiniCPM-o 2.6	GPT-4o-mini	Qwen2-VL-7B	Qwen2.5-VL-7B
CC_OCR	57.7			61.6	**77.8**
MMStar	62.8			60.7	**63.9**
MMBench-V1.1-En$_{test}$	79.4	78.0	76.0	80.7	**82.6**
MMT-Bench$_{test}$	-	-	-	**63.7**	63.6
MMStar	**61.5**	57.5	54.8	60.7	63.9
MMVet$_{GPT-4-Turbo}$	54.2	60.0	66.9	62.0	**67.1**
HallBench$_{avg}$	45.2	48.1	46.1	50.6	**52.9**
MathVista$_{testmini}$	58.3	60.6	52.4	58.2	**68.2**
MathVision	-	-	-	16.3	**25.07**

Video Benchmarks

Benchmark	Qwen2-VL-7B	Qwen2.5-VL-7B
MVBench	67.0	**69.6**
PerceptionTest$_{test}$	66.9	**70.5**
Video-MME$_{wo/w\ subs}$	63.3/69.0	**65.1**/**71.6**
LVBench		45.3
LongVideoBench		54.7
MMBench-Video	1.44	1.79
TempCompass		71.7
MLVU		70.2
CharadesSTA/mIoU	43.6	

Agent benchmark

Benchmarks	Qwen2.5-VL-7B
ScreenSpot	84.7
ScreenSpot Pro	29.0
AITZ_EM	81.9
Android Control High_EM	60.1
Android Control Low_EM	93.7
AndroidWorld_SR	25.5
MobileMiniWob++_SR	91.4

Requirements

The code of Qwen2.5-VL has been in the latest Hugging face transformers and we advise you to build from source with command:

```
pip install git+https://github.com/huggingface/transformers accelerate
```

or you might encounter the following error:

```
KeyError: 'qwen2_5_vl'
```

Quickstart

Below, we provide simple examples to show how to use Qwen2.5-VL with 🤖 ModelScope and 🤗 Transformers.

The code of Qwen2.5-VL has been in the latest Hugging face transformers and we advise you to build from source with command:

```
pip install git+https://github.com/huggingface/transformers accelerate
```

or you might encounter the following error:

```
KeyError: 'qwen2_5_vl'
```

We offer a toolkit to help you handle various types of visual input more conveniently, as if you were using an API. This includes base64, URLs, and interleaved images and videos. You can install it using the following command:

```
# It's highly recommanded to use `[decord]` feature for faster video loading.
pip install qwen-vl-utils[decord]==0.0.8
```

If you are not using Linux, you might not be able to install `decord` from PyPI. In that case, you can use `pip install qwen-vl-utils` which will fall back to using torchvision for video processing. However, you can still install decord from source to get decord used when loading video.

Using 🤗 Transformers to Chat

Here we show a code snippet to show you how to use the chat model with `transformers` and `qwen_vl_utils`:

```
from transformers import Qwen2_5_VLForConditionalGeneration, AutoTokenizer, AutoProcessor
from qwen_vl_utils import process_vision_info

# default: Load the model on the available device(s)
model = Qwen2_5_VLForConditionalGeneration.from_pretrained(
    "Qwen/Qwen2.5-VL-7B-Instruct", torch_dtype="auto", device_map="auto"
)

# We recommend enabling flash_attention_2 for better acceleration and memory saving, especially i
# model = Qwen2_5_VLForConditionalGeneration.from_pretrained(
#     "Qwen/Qwen2.5-VL-7B-Instruct",
#     torch_dtype=torch.bfloat16,
#     attn_implementation="flash_attention_2",
#     device_map="auto",
# )

# default processer
processor = AutoProcessor.from_pretrained("Qwen/Qwen2.5-VL-7B-Instruct")

# The default range for the number of visual tokens per image in the model is 4-16384.
# You can set min_pixels and max_pixels according to your needs, such as a token range of 256-128
# min_pixels = 256*28*28
# max_pixels = 1280*28*28
# processor = AutoProcessor.from_pretrained("Qwen/Qwen2.5-VL-7B-Instruct", min_pixels=min_pixels,

messages = [
    {
        "role": "user",
        "content": [
            {
```

```
                "type": "image",
                "image": "https://qianwen-res.oss-cn-beijing.aliyuncs.com/Qwen-VL/assets/demo.jpe
            },
            {"type": "text", "text": "Describe this image."},
        ],
    }
]

# Preparation for inference
text = processor.apply_chat_template(
    messages, tokenize=False, add_generation_prompt=True
)
image_inputs, video_inputs = process_vision_info(messages)
inputs = processor(
    text=[text],
    images=image_inputs,
    videos=video_inputs,
    padding=True,
    return_tensors="pt",
)
inputs = inputs.to("cuda")

# Inference: Generation of the output
generated_ids = model.generate(**inputs, max_new_tokens=128)
generated_ids_trimmed = [
    out_ids[len(in_ids) :] for in_ids, out_ids in zip(inputs.input_ids, generated_ids)
]
output_text = processor.batch_decode(
    generated_ids_trimmed, skip_special_tokens=True, clean_up_tokenization_spaces=False
)
print(output_text)
```

▶ Multi image inference

▶ Video inference

▶ Batch inference

🤖 ModelScope

We strongly advise users especially those in mainland China to use ModelScope. `snapshot_download` can help you solve issues concerning downloading checkpoints.

More Usage Tips

For input images, we support local files, base64, and URLs. For videos, we currently only support local files.

```
# You can directly insert a local file path, a URL, or a base64-encoded image into the position w
## Local file path
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image", "image": "file:///path/to/your/image.jpg"},
            {"type": "text", "text": "Describe this image."},
        ],
    }
]
## Image URL
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image", "image": "http://path/to/your/image.jpg"},
            {"type": "text", "text": "Describe this image."},
        ],
    }
]
## Base64 encoded image
messages = [
    {
        "role": "user",
        "content": [
            {"type": "image", "image": "data:image;base64,/9j/..."},
            {"type": "text", "text": "Describe this image."},
        ],
    }
]
```

Image Resolution for performance boost

The model supports a wide range of resolution inputs. By default, it uses the native resolution for input, but higher resolutions can enhance performance at the cost of more computation. Users can set the minimum and maximum number of pixels to achieve an optimal configuration for their needs, such as a token count range of 256-1280, to balance speed and memory usage.

```
min_pixels = 256 * 28 * 28
max_pixels = 1280 * 28 * 28
processor = AutoProcessor.from_pretrained(
    "Qwen/Qwen2.5-VL-7B-Instruct", min_pixels=min_pixels, max_pixels=max_pixels
)
```

Besides, We provide two methods for fine-grained control over the image size input to the model:

1. Define min_pixels and max_pixels: Images will be resized to maintain their aspect ratio within the range of min_pixels and max_pixels.

2. Specify exact dimensions: Directly set `resized_height` and `resized_width`. These values will be rounded to the nearest multiple of 28.

```
# min_pixels and max_pixels
messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image",
                "image": "file:///path/to/your/image.jpg",
                "resized_height": 280,
                "resized_width": 420,
            },
            {"type": "text", "text": "Describe this image."},
        ],
    }
]
# resized_height and resized_width
messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image",
                "image": "file:///path/to/your/image.jpg",
                "min_pixels": 50176,
                "max_pixels": 50176,
            },
            {"type": "text", "text": "Describe this image."},
        ],
    }
]
```

Processing Long Texts

The current `config.json` is set for context length up to 32,768 tokens. To handle extensive inputs exceeding 32,768 tokens, we utilize YaRN, a technique for enhancing model length extrapolation, ensuring optimal performance on lengthy texts.

For supported frameworks, you could add the following to `config.json` to enable YaRN:

{ ..., "type": "yarn", "mrope_section": [16, 24, 24], "factor": 4, "original_max_position_embeddings": 32768 }

However, it should be noted that this method has a significant impact on the performance of temporal and spatial localization tasks, and is therefore not recommended for use.

At the same time, for long video inputs, since MRoPE itself is more economical with ids, the max_position_embeddings can be directly modified to a larger value, such as 64k.

Citation

If you find our work helpful, feel free to give us a cite.

```
@misc{qwen2.5-VL,
    title = {Qwen2.5-VL},
    url = {https://qwenlm.github.io/blog/qwen2.5-vl/},
    author = {Qwen Team},
    month = {January},
    year = {2025}
}

@article{Qwen2VL,
  title={Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution},
  author={Wang, Peng and Bai, Shuai and Tan, Sinan and Wang, Shijie and Fan, Zhihao and Bai, Jinz
  journal={arXiv preprint arXiv:2409.12191},
  year={2024}
}

@article{Qwen-VL,
  title={Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading
  author={Bai, Jinze and Bai, Shuai and Yang, Shusheng and Wang, Shijie and Tan, Sinan and Wang,
  journal={arXiv preprint arXiv:2308.12966},
  year={2023}
}
```