Vision

Learn how to use GPT-4 to understand images

Introduction

GPT-4 Turbo with Vision allows the model to take in images and answer questions about them. Historically, language model systems have been limited by taking in a single input modality, text. For many use cases, this constrained the areas where models like GPT-4 could be used. Previously, the model has sometimes been referred to as GPT-4V or `gpt-4-vision-preview` in the API. Plese note that the Assistants API does not currently support image inputs.

Quick start

Images are made available to the model in two main ways: by passing a link to the image or by passing the base64 encoded image directly in the request. Images can be passed in the `user`, `system` and `assistant` messages. Currently we don't support images in the first `system` message but this may change in the future.

What's in this image? python

```python
from openai import OpenAI

client = OpenAI()

response = client.chat.completions.create(
  model="gpt-4-turbo",
  messages=[
    {
      "role": "user",
      "content": [
        {"type": "text", "text": "What’s in this image?"},
        {
          "type": "image_url",
          "image_url": {
            "url": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/Gfp-wisco
          },
        },
      ],
    }
  ],
  max_tokens=300,
)
```

```
23
24 print(response.choices[0])
```

The model is best at answering general questions about what is present in the images. While it does understand the relationship between objects in images, it is not yet optimized to answer detailed questions about the location of certain objects in an image. For example, you can ask it what color a car is or what some ideas for dinner might be based on what is in you fridge, but if you show it an image of a room and ask it where the chair is, it may not answer the question correctly.

It is important to keep in mind the limitations of the model as you explore what use-cases visual understanding can be applied to.

Video understanding with vision

Learn how to use use GPT-4 with Vision to understand videos in the OpenAI Cookbook

Uploading base 64 encoded images

If you have an image or set of images locally, you can pass those to the model in base 64 encoded format, here is an example of this in action:

```python
import base64
import requests

# OpenAI API Key
api_key = "YOUR_OPENAI_API_KEY"

# Function to encode the image
def encode_image(image_path):
  with open(image_path, "rb") as image_file:
    return base64.b64encode(image_file.read()).decode('utf-8')

# Path to your image
image_path = "path_to_your_image.jpg"

# Getting the base64 string
base64_image = encode_image(image_path)

headers = {
  "Content-Type": "application/json",
```

```
  "Authorization": f"Bearer {api_key}"
}

payload = {
  "model": "gpt-4-turbo",
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "What's in this image?"
        },
        {
          "type": "image_url",
          "image_url": {
            "url": f"{base64_image}"
          }
        }
      ]
    }
  ],
  "max_tokens": 300
}

response = requests.post("https://api.openai.com/v1/chat/completions", head

print(response.json())
```

Multiple image inputs

The Chat Completions API is capable of taking in and processing multiple image inputs in both base64 encoded format or as an image URL. The model will process each image and use the information from all of them to answer the question.

Multiple image inputs — python

```
from openai import OpenAI

```

```
client = OpenAI()
response = client.chat.completions.create(
  model="gpt-4-turbo",
  messages=[
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "What are in these images? Is there any difference between them?",
        },
        {
          "type": "image_url",
          "image_url": {
            "url": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/Gfp-wisco
          },
        },
        {
          "type": "image_url",
          "image_url": {
            "url": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/Gfp-wisco
          },
        },
      ],
    }
  ],
  max_tokens=300,
)
print(response.choices[0])
```

Here the model is shown two copies of the same image and can answer questions about both or each

Low or high fidelity image understanding

By controlling the `detail` parameter, which has three options, `low`, `high`, or `auto`, you have control over how the model processes the image and generates its textual understanding. By default, the model will use the `auto` setting which will look at the image input size and decide if it should use the `low` or `high` setting.

- `low` will enable the "low res" mode. The model will receive a low-res 512px x 512px version of the image, and represent the image with a budget of 65 tokens. This allows the API to return faster responses and consume fewer input tokens for use cases that do not require high detail.

- `high` will enable "high res" mode, which first allows the model to see the low res image and then creates detailed crops of input images as 512px squares based on the input image size. Each of the detailed crops uses twice the token budget (65 tokens) for a total of 129 tokens.

Choosing the detail level

python

```python
from openai import OpenAI

client = OpenAI()

response = client.chat.completions.create(
  model="gpt-4-turbo",
  messages=[
    {
      "role": "user",
      "content": [
        {"type": "text", "text": "What's in this image?"},
        {
          "type": "image_url",
          "image_url": {
            "url": "https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/Gfp-wisco
            "detail": "high"
          },
        },
      ],
    }
  ],
  max_tokens=300,
)

print(response.choices[0].message.content)
```

Managing images

The Chat Completions API, unlike the Assistants API, is not stateful. That means you have to manage the messages (including images) you pass to the model yourself. If you want to pass the same image to the model multiple times, you will have to pass the image each time you make a request to the API.

For long running conversations, we suggest passing images via URL's instead of base64. The latency of the model can also be improved by downsizing your images ahead of time to be less than the maximum size they are expected them to be. For low res mode, we expect a 512px x 512px image. For high res mode, the short side of the image should be less than 768px and the long side should be less than 2,000px.

After an image has been processed by the model, it is deleted from OpenAI servers and not retained. We do not use data uploaded via the OpenAI API to train our models.

Limitations

While GPT-4 with vision is powerful and can be used in many situations, it is important to understand the limitations of the model. Here are some of the limitations we are aware of:

- Medical images: The model is not suitable for interpreting specialized medical images like CT scans and shouldn't be used for medical advice.
- Non-English: The model may not perform optimally when handling images with text of non-Latin alphabets, such as Japanese or Korean.
- Small text: Enlarge text within the image to improve readability, but avoid cropping important details.
- Rotation: The model may misinterpret rotated / upside-down text or images.
- Visual elements: The model may struggle to understand graphs or text where colors or styles like solid, dashed, or dotted lines vary.
- Spatial reasoning: The model struggles with tasks requiring precise spatial localization, such as identifying chess positions.
- Accuracy: The model may generate incorrect descriptions or captions in certain scenarios.
- Image shape: The model struggles with panoramic and fisheye images.
- Metadata and resizing: The model doesn't process original file names or metadata, and images are resized before analysis, affecting their original dimensions.
- Counting: May give approximate counts for objects in images.
- CAPTCHAS: For safety reasons, we have implemented a system to block the submission of CAPTCHAs.

Calculating costs

Image inputs are metered and charged in tokens, just as text inputs are. The token cost of a given image is determined by two factors: its size, and the `detail` option on each image_url block. All images with `detail: low` cost 85 tokens each. `detail: high` images are first scaled to fit within a 2048 × 2048 square, maintaining their aspect ratio. Then, they are scaled such that the shortest side of the image is 768px long. Finally, we count how many 512px squares the image consists of. Each of those squares costs **170 tokens**. Another **85 tokens** are always added to the final total.

Here are some examples demonstrating the above.

- A 1024 × 1024 square image in `detail: high` mode costs 765 tokens

    - 1024 is less than 2048, so there is no initial resize.
    - The shortest side is 1024, so we scale the image down to 768 × 768.
    - 4 512px square tiles are needed to represent the image, so the final token cost is `170 * 4 + 85 = 765`.
- A 2048 × 4096 image in `detail: high` mode costs 1105 tokens
    - We scale down the image to 1024 × 2048 to fit within the 2048 square.
    - The shortest side is 1024, so we further scale down to 768 × 1536.
    - 6 512px tiles are needed, so the final token cost is `170 * 6 + 85 = 1105`.
- A 4096 × 8192 image in `detail: low` most costs 85 tokens
    - Regardless of input size, low detail images are a fixed cost.

FAQ

Can I fine-tune the image capabilities in `gpt-4`?

No, we do not support fine-tuning the image capabilities of `gpt-4` at this time.

Can I use `gpt-4` to generate images?

No, you can use `dall-e-3` to generate images and `gpt-4-turbo` to understand images.

What type of files can I upload?

We currently support PNG (.png), JPEG (.jpeg and .jpg), WEBP (.webp), and non-animated GIF (.gif).

Is there a limit to the size of the image I can upload?

Yes, we restrict image uploads to 20MB per image.

Can I delete an image I uploaded?

No, we will delete the image for you automatically after it has been processed by the model.

Where can I learn more about the considerations of GPT-4 with Vision?

You can find details about our evaluations, preparation, and mitigation work in the GPT-4 with Vision system card.

We have further implemented a system to block the submission of CAPTCHAs.

How do rate limits for GPT-4 with Vision work?

We process images at the token level, so each image we process counts towards your tokens per minute (TPM) limit. See the calculating costs section for details on the formula used to determine token count per image.

Can GPT-4 with Vision understand image metadata?

No, the model does not receive image metadata.

What happens if my image is unclear?

If an image is ambiguous or unclear, the model will do its best to interpret it. However, the results may be less accurate. A good rule of thumb is that if an average human cannot see the info in an image at the resolutions used in low/high res mode, then the model cannot either.