ChatGPT

Using Vision in the OpenAI API (October 2025)

OpenAI's latest models support **multimodal vision** capabilities, enabling you to feed in images (e.g. document scans) and get rich text outputs like structured JSON. Below, we outline the state-of-the-art methods (as of Oct 2025) for using vision via the OpenAI API, clarify the role of the `gpt-image-1` model, and show how to call the API with image inputs for tasks like document analysis.

Vision-Capable Models in 2025

OpenAI's multimodal GPT models can analyze images and respond in text. Key vision-enabled models include:

- **GPT-4 Turbo with Vision:** An enhanced GPT-4 model (released April 2024) that introduced image recognition via the API [1] [2]. It offers faster responses and large context windows (up to 128k tokens) and allows JSON/function-call outputs for structured tasks [2].
- **GPT-4 *"omni"* (GPT-4o):** Introduced May 2024, GPT-4o is a multimodal model ("o" for *omni*) that natively accepts **text + image inputs** and produces text outputs [3]. It matches GPT-4 Turbo's language performance while *significantly improving vision understanding*, and it's optimized for speed and cost (about 50% cheaper on the API) [3]. A smaller **GPT-4o-mini** variant offers cost-efficient multimodal reasoning with similar image input support [4] [5].
- **GPT-4.5 and GPT-4.1 series:** Incremental improvements on GPT-4 with vision (as referenced by Azure) – these also support image inputs [6].
- **GPT-5 series:** OpenAI's newest flagship (launched mid-2025) is **multimodal** out-of-the-box. GPT-5 can "*reason more accurately over images and other non-text inputs*" [7], achieving state-of-the-art visual understanding on benchmarks. In other words, the GPT-5 model **"sees"** images with even greater accuracy than GPT-4o. By October 2025, the GPT-5 model (and cost-efficient variants like GPT-5 mini/nano) supports **image input** through the API, with very large context windows (e.g. 400K tokens) for combining lengthy text and images [8].

Availability: Vision features are generally available in the ChatGPT UI (e.g. GPT-5 powered ChatGPT can accept image uploads). For developers, OpenAI's Chat Completion API supports these vision models provided you have access to them. According to Microsoft's documentation, *"vision-enabled chat models"* currently include the GPT-5 series, GPT-4.1/4.5, GPT-4o series, and GPT-4 Turbo with Vision [6]. Using these models via the API, you can send images in your prompt and receive the model's analysis as text.

GPT-Image-1 vs. GPT Vision Models

It's important to distinguish `gpt-image-1` from the chat models above:

- `gpt-image-1` is **OpenAI's image generation model** (text-to-image). Announced in April 2025, it's the API counterpart of ChatGPT's image generator [9]. This model is *"the natively multimodal model that powers [image generation in ChatGPT]"* and is geared for creating images from prompts [10]. It excels at producing diverse styles of images, following prompts, and even rendering text in images

[11]. In short, `gpt-image-1` is used when you want the AI *to generate an image output*, not to analyze an input image.

- **Vision-enabled GPT-4/GPT-5** (GPT-4 Vision, GPT-4o, GPT-5, etc.) are **chat/completion models** that take image inputs and produce **textual analysis**. These are the models you'd use to *read and understand document images* and output JSON or text summaries.

So, for your use-case (parsing PDFs converted to images and extracting structured data), **you would NOT use** `gpt-image-1`. Instead, you'd use a **chat completion model with vision** support (such as GPT-4o or GPT-5). These models are designed to interpret images (including reading text from them and understanding their content) and respond with text.

Using the API with Image Inputs

OpenAI's Chat Completion API allows image inputs by formatting the message content as an *array* of parts, including text and images. The general approach is:

1. **Choose a vision-capable model** in the API call (e.g. `"model": "gpt-4o"` or `"gpt-5"` depending on availability).
2. **Structure your prompt message** as an array of content parts, mixing your text prompt and the image. For example, a JSON payload for the user message might look like:

```
{
  "role": "user",
  "content": [
    { "type": "text", "text": "Describe the document and extract key fields as
JSON:" },
    {
      "type": "image_url",
      "image_url": { "url": "<PUBLIC_IMAGE_URL or DATA_URL>" }
    }
  ]
}
```

OpenAI's API will accept either an **image URL** (must be publicly accessible or a presigned URL) or a **base64 data URL** as the image content [12]. In either case, the `content` becomes a list where you first provide any textual instruction and then include the image. The model will "see" the image and the prompt together as the user input.

Example: Using the official Python library, you might do:

```
response = openai.ChatCompletion.create(
  model="gpt-4o",
  messages=[
    {"role": "user", "content": [
        {"type": "text", "text": "What does this document say? Give output as
```

```
JSON."},
        {"type": "image_url", "image_url": {"url": "https://example.com/
mydoc_page1.png"}}
    ]}
  ],
  max_tokens=1000
)
print(response.choices[0].message.content)
```

In the above code, we set the model to GPT-4o (which supports vision) and provide a user message consisting of text and an image. **GPT-4o** will then analyze *"mydoc_page1.png"* and respond accordingly. This is exactly how you'd stream PDF pages (converted to images) into the model. In an MLJAR example, a similar call was made to ask GPT-4o *"What is in the image?"* by providing a URL, and the API successfully returned a description [13] [14].

Note: If you have the image data locally, you can base64-encode it and send it as a data URL string in place of a live URL [12]. There is also an alternate content type `"input_image"` for directly uploading image data via certain endpoints (or the Azure OpenAI service) [15], but using `image_url` with a data URL achieves the same result for OpenAI's API.

Limits: You can include **up to 10 images** in a single request (as separate content parts) [16]. This is useful if you want the model to consider multiple pages or visuals together. Each image counts toward the model's context length (they are tokenized internally), so very large images or many images will consume more "image tokens" (OpenAI charges image inputs per token, similar to text [17]).

JSON Outputs and Post-Processing

Since your goal is to produce structured JSON from the document, you can simply instruct the model accordingly. The latest GPT-4/GPT-5 models are quite adept at following format instructions (the ChatGPT UI's success you observed confirms this). You might prompt: *"Extract the following fields from the contract and return them in a JSON object with keys X, Y, Z."*

Furthermore, OpenAI introduced **function calling** in the Chat Completion API, which can enforce structured outputs. You can define a function with a schema corresponding to your desired JSON format and ask the model to `call` that function with parsed arguments. The model will then output JSON that matches the schema. This is the same mechanism ChatGPT uses to reliably produce JSON or code when needed. (For instance, GPT-4 Turbo with Vision was shown generating JSON code snippets to automate actions via function calls [2].) Using function calling, the model's response will come in a machine-readable JSON form, reducing the chance of formatting errors.

Summary of Best Practices

- **Use the correct model:** For vision analysis, use a multimodal model like GPT-4o or GPT-5 – *not* the `gpt-image-1` generation model [10].
- **Format your API request properly:** Send a chat completion with the image included as a content part of the user message. The OpenAI API expects an array of `{type: ..., ...}` objects for

mixed media input [12]. Ensure the image is accessible (URL or base64 data) and consider the 10-image limit per request [16].
- **Leverage model capability:** These models can read printed and handwritten text, interpret charts or forms, and understand the layout of documents. GPT-5 in particular has strong visual reasoning skills [7], which should handle complex documents like adoption agreements well.
- **Guide the output:** In your prompt or system message, clearly state that the answer should be JSON and describe the desired keys/structure. Optionally use function calling to enforce the JSON format.
- **Stay updated:** OpenAI's documentation (see the *Vision* API guide and model cards) will have the latest details on input formatting and model availability. As of Oct 2025, vision AI is a fast-evolving area – GPT-5's multimodal abilities are the new state of the art, building on the robust foundation of GPT-4's vision model [7].

By following these practices, you can stream PDF pages as images into the OpenAI API and have the model parse and output structured JSON data about those documents. The **bottom line**: OpenAI's native API now fully supports image inputs via chat completions, using advanced multimodal models to "see" the content and respond with the analysis you need. With GPT-4o and GPT-5, this capability is available for developers to integrate into their own tools [6] [13].

Sources

- OpenAI Announcement – *Introducing our latest image generation model in the API (gpt-image-1)* [9] [10]
- OpenAI Milestone – *Hello GPT-4o* (Multimodal model intro, May 2024) [3]
- Microsoft Learn – *Use vision-enabled chat models (Azure OpenAI)* [6] [18] [16]
- MLJAR Example – *OpenAI vision with URL images in Python* (GPT-4o usage) [13] [14]
- OpenAI Blog – *GPT-5 is here* (Multimodal capabilities of GPT-5) [7]
- *Artificial Intelligence News – GPT-4 Turbo with Vision API generally available* (API JSON outputs & use cases) [2] [19]

---

[1] [2] [19] OpenAI makes GPT-4 Turbo with Vision API generally available
https://www.artificialintelligence-news.com/news/openai-gpt-4-turbo-with-vision-api-generally-available/

[3] Hello GPT-4o | OpenAI
https://openai.com/index/hello-gpt-4o/

[4] [5] node.js - BadRequestError: 400 Invalid content type. image_url is only supported by certain models - Stack Overflow
https://stackoverflow.com/questions/78535109/badrequesterror-400-invalid-content-type-image-url-is-only-supported-by-certai

[6] [12] [16] [18] How to use vision-enabled chat models - Azure OpenAI in Azure AI Foundry Models | Microsoft Learn
https://learn.microsoft.com/en-us/azure/ai-foundry/openai/how-to/gpt-with-vision

[7] Introducing GPT-5 | OpenAI
https://openai.com/index/introducing-gpt-5/

[8] GPT-5 is here | OpenAI
https://openai.com/gpt-5/

9 10 11 17 Introducing our latest image generation model in the API | OpenAI
https://openai.com/index/image-generation-api/

13 14 OpenAI vision with URL images in Python
https://mljar.com/notebooks/openai-vision-url-image/

15 How to upload image to GPT-4o using Response API
https://community.openai.com/t/how-to-upload-image-to-gpt-4o-using-response-api/1333438