

# Vision

Claude's vision capabilities allow it to understand and analyze images, opening up exciting possibilities for multimodal interaction.

 Copy page 

This guide describes how to work with images in Claude, including best practices, code examples, and limitations to keep in mind.

## How to use vision

Use Claude's vision capabilities via:

- [claude.ai](#). Upload an image like you would a file, or drag and drop an image directly into the chat window.
- The [Console Workbench](#). A button to add images appears at the top right of every User message block.
- **API request**. See the examples in this guide.

## Before you upload

### Basics and Limits


You can include multiple images in a single request (up to 20 for [claude.ai](#) and 100 for API requests). Claude will analyze all provided images when formulating its response. This can be helpful for comparing or contrasting images.

If you submit an image larger than 8000x8000 px, it will be rejected. If you submit more than 20 images in one API request, this limit is 2000x2000 px.

## Evaluate image size

For optimal performance, we recommend resizing images before uploading if they are too large. If your image's long edge is more than 1568 pixels, or your image is more than ~1,600 tokens, it will first be scaled down, preserving aspect ratio, until it's within the size limits.

If your input image is too large and needs to be resized, it will increase latency of time-to-first-token, without giving you any additional model performance. Very small images under 200 pixels on any given edge may degrade performance.

 To improve time-to-first-token, we recommend resizing images to no more than 1.15 megapixels (and within 1568 pixels in both dimensions).

Here is a table of maximum image sizes accepted by our API that will not be resized for common aspect ratios. With Claude Sonnet 4.5, these images use approximately 1,600 tokens and around \$4.80/1K images.

Aspect ratio	Image size
1:1	1092x1092 px
3:4	951x1268 px
2:3	896x1344 px
9:16	819x1456 px
1:2	784x1568 px

## Calculate image costs

Each image you include in a request to Claude counts towards your token usage. To calculate the approximate cost, multiply the approximate number of image tokens by the per-token price of the model you're using.

If your image does not need to be resized, you can estimate the number of tokens used through this algorithm: `tokens = (width px * height px)/750`

Here are examples of approximate tokenization and costs for different image sizes within our API's size constraints based on Claude Sonnet 4.5 per-token price of \$3 per million

## Claude Docs

image size	# of tokens	Cost / image	Cost / 1K images
200x200 px(0.04 megapixels)	~54	~\$0.00016	~\$0.16
1000x1000 px(1 megapixel)	~1334	~\$0.004	~\$4.00
1092x1092 px(1.19 megapixels)	~1590	~\$0.0048	~\$4.80

## Ensuring image quality

When providing images to Claude, keep the following in mind for best results:

- **Image format:** Use a supported image format: JPEG, PNG, GIF, or WebP.
- **Image clarity:** Ensure images are clear and not too blurry or pixelated.
- **Text:** If the image contains important text, make sure it's legible and not too small. Avoid cropping out key visual context just to enlarge the text.

## Prompt examples

Many of the [prompting techniques](#) that work well for text-based interactions with Claude can also be applied to image-based prompts.

These examples demonstrate best practice prompt structures involving images.



Just as with document-query placement, Claude works best when images come before text. Images placed after text or interpolated with text will still perform well, but if your use case allows it, we recommend an image-then-text structure.

## About the prompt examples

The following examples demonstrate how to use Claude's vision capabilities using various programming languages and approaches. You can provide images to Claude in three ways:

1. As a base64-encoded image in `image` content blocks
2. As a URL reference to an image hosted online
3. Using the Files API (upload once, use multiple times)

The base64 example prompts use these variables:



```
# For URL-based images, you can use the URL directly in your JSON request

# For base64-encoded images, you need to first encode the image
# Example of how to encode an image to base64 in bash:
BASE64_IMAGE_DATA=$(curl -s "https://upload.wikimedia.org/wikipedia/common

# The encoded data can now be used in your API calls
```

Below are examples of how to include images in a Messages API request using base64-encoded images and URL references:

## Base64-encoded image example

Shell ▾





```
-H "anthropic-version: 2023-06-01" \  
-H "content-type: application/json" \  
-d '{  
  "model": "claude-sonnet-4-5",  
  "max_tokens": 1024,  
  "messages": [  
    {  
      "role": "user",  
      "content": [  
        {  
          "type": "image",  
          "source": {  
            "type": "base64",  
            "media_type": "image/jpeg",  
            "data": ""$BASE64_IMAGE_DATA""  
          }  
        },  
        {  
          "type": "text",  
          "text": "Describe this image."  
        }  
      ]  
    }  
  ]  
}'
```

## URL-based image example

Shell ▾





```
-H "anthropic-version: 2023-06-01" \  
-H "content-type: application/json" \  
-d '{  
  "model": "claude-sonnet-4-5",  
  "max_tokens": 1024,  
  "messages": [  
    {  
      "role": "user",  
      "content": [  
        {  
          "type": "image",  
          "source": {  
            "type": "url",  
            "url": "https://upload.wikimedia.org/wikipedia/commons/a/a7/Camp  
        }  
      },  
      {  
        "type": "text",  
        "text": "Describe this image."  
      }  
    ]  
  }  
]
```

## Files API image example

For images you'll use repeatedly or when you want to avoid encoding overhead, use the [Files API](#):

Shell ▾





```
-H "x-api-key: $ANTHROPIC_API_KEY" \  
-H "anthropic-version: 2023-06-01" \  
-H "anthropic-beta: files-api-2025-04-14" \  
-F "file=@image.jpg"
```

# Then use the returned file\_id in your message

```
curl https://api.anthropic.com/v1/messages \  
-H "x-api-key: $ANTHROPIC_API_KEY" \  
-H "anthropic-version: 2023-06-01" \  
-H "anthropic-beta: files-api-2025-04-14" \  
-H "content-type: application/json" \  
-d '{  
  "model": "claude-sonnet-4-5",  
  "max_tokens": 1024,  
  "messages": [  
    {  
      "role": "user",  
      "content": [  
        {  
          "type": "image",  
          "source": {  
            "type": "file",  
            "file_id": "file_abc123"  
          }  
        },  
        {  
          "type": "text",  
          "text": "Describe this image."  
        }  
      ]  
    }  
  ]  
}'
```

See [Messages API examples](#) for more example code and parameter details.

> **Example: One image**

> **Example: Multiple images**

> **Example: Multiple images with a system prompt**

## Limitations

While Claude's image understanding capabilities are cutting-edge, there are some limitations to be aware of:

- **People identification:** Claude cannot be used to identify (i.e., name) people in images and will refuse to do so.
- **Accuracy:** Claude may hallucinate or make mistakes when interpreting low-quality, rotated, or very small images under 200 pixels.
- **Spatial reasoning:** Claude's spatial reasoning abilities are limited. It may struggle with tasks requiring precise localization or layouts, like reading an analog clock face or describing exact positions of chess pieces.
- **Counting:** Claude can give approximate counts of objects in an image but may not always be precisely accurate, especially with large numbers of small objects.
- **AI generated images:** Claude does not know if an image is AI-generated and may be incorrect if asked. Do not rely on it to detect fake or synthetic images.
- **Inappropriate content:** Claude will not process inappropriate or explicit images that violate our Acceptable Use Policy.
- **Healthcare applications:** While Claude can analyze general medical images, it is not designed to interpret complex diagnostic scans such as CTs or MRIs. Claude's outputs should not be considered a substitute for professional medical advice or diagnosis.

Always carefully review and verify Claude's image interpretations, especially for high-stakes use cases. Do not use Claude for tasks requiring perfect precision or sensitive image analysis without human oversight.

## FAQ

- > **What image file types does Claude support?**





- › **Is there a limit to the image file size I can upload?**
- › **How many images can I include in one request?**
- › **Does Claude read image metadata?**
- › **Can I delete images I've uploaded?**
- › **Where can I find details on data privacy for image uploads?**
- › **What if Claude's image interpretation seems wrong?**
- › **Can Claude generate or edit images?**

## Dive deeper into vision

Ready to start building with images using Claude? Here are a few helpful resources:

- [Multimodal cookbook](#): This cookbook has tips on [getting started with images](#) and [best practice techniques](#) to ensure the highest quality performance with images. See how you can effectively prompt Claude with images to carry out tasks such as [interpreting and analyzing charts](#) or [extracting content from forms](#).
- [API reference](#): Visit our documentation for the Messages API, including example [API calls involving images](#).

If you have any other questions, feel free to reach out to our [support team](#). You can also join our [developer community](#) to connect with other creators and get help from Anthropic experts.





---

## Solutions

AI agents

Code modernization

Coding

Customer support

Education

Financial services

Government

Life sciences

Partners

Amazon Bedrock

Google Cloud's Vertex AI

## Learn

Blog

Catalog

Courses

Use cases

Connectors

Customer stories

Engineering at Anthropic

Events

Powered by Claude

Service partners

Startups program

Company

Anthropic

Careers

Economic Futures

Research

News

Responsible Scaling Policy

Security and compliance

Transparency

Help and security

Availability

Status

Support

Discord

Terms and policies

Privacy policy

Responsible disclosure policy

Terms of service: Commercial

Terms of service: Consumer

Usage policy

