

# Image Model Feasibility Check

Prove by hand that the image model can do the thing before you write a single line wiring it up.

---

## 01 When to use this

Use this before you build anything on top of an image model. Ten minutes clicking around in the model's own interface tells you whether the pipeline is possible at all. Skip it and you can spend a day automating something the model was never going to do.

## 02 The prompt

STEP 1, do by hand before prompting Claude:

Open `<IMAGE_TOOL>` directly. Upload `<REFERENCE_AD_IMAGE>` and `<PRODUCT_PHOTO>`.

Ask it to apply the product to that ad format. Look at the result. If it is not acceptable, stop here. Do not build anything.

STEP 2, the wiring prompt:

I want to use `<IMAGE_MODEL_NAME>` in this pipeline. Before you write any code:

1. Go read the official API documentation for `<IMAGE_MODEL_NAME>`. Fetch every endpoint, function, and parameter I would need to steer this model: image input, reference images, size, quality, output format, and anything else that controls the result.
2. Confirm my API key is available in the environment. It is set as `<ENV_VAR_NAME>`, e.g. `OPENAI_API_KEY`. You will be able to use it but not read it back, that is expected.
3. Run exactly ONE live test generation using `<REFERENCE_AD_IMAGE>` and `<PRODUCT_PHOTO>`. One. Do not batch anything yet.
4. Show me the output and the exact parameters you used.

Use `<IMAGE_MODEL_NAME>` specifically. Do not fall back to an older version of the model. If the endpoint you want is not in the docs, tell me instead of guessing.

## 03 What it is actually doing

There are two prompts here and the first one is not a prompt at all. It is you, clicking, in the model's own interface.

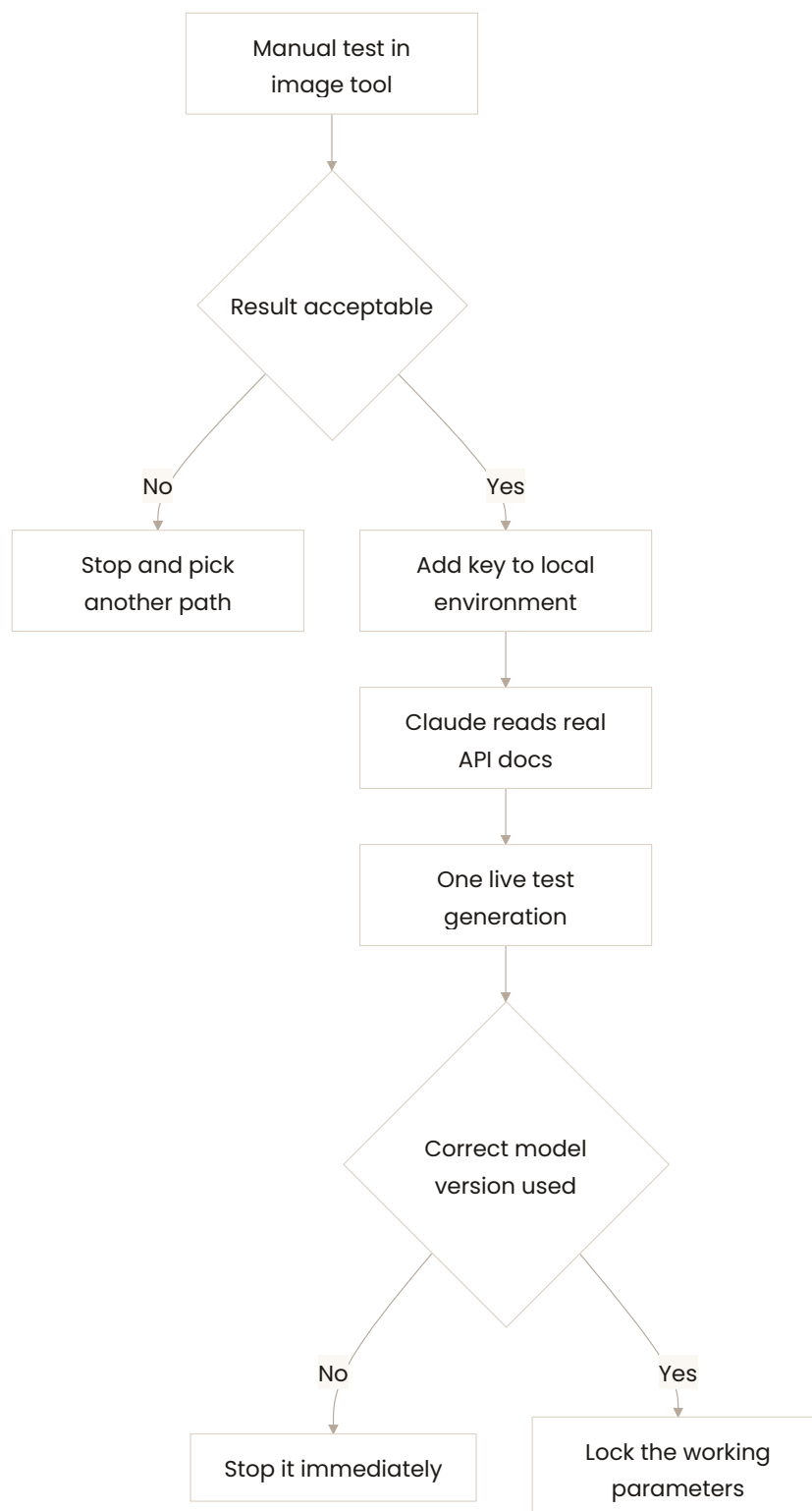
Nick opened the image tool directly, uploaded a reference ad and a product photo, and asked it to put the product into that ad layout. He looked at the result. Only after seeing something acceptable did he go anywhere near wiring it up. That is the whole feasibility check. It costs ten minutes and it can save a day.

The second prompt is the wiring, and it has a specific shape. You tell Claude to go read the real API documentation first. Not remember it, read it. Models have stale memories of APIs and will confidently call an endpoint that no longer exists. Then one live test generation. One. You are checking the plumbing, not making ads yet.

On keys: the API key goes in the local environment panel as `OPENAI_API_KEY` equals your key. Claude can use that key but cannot read it back to you. That is on purpose. It is a defense against prompt injection, where a malicious web page tells the model to reveal your secrets. The model cannot leak what it cannot see.

One more thing Nick did that is worth copying. He watched the model start reaching for the previous version of the image model and stopped it immediately. Correct errors the moment you see them. Letting a wrong turn run burns tokens and produces a pile of output you then have to unwind.

## 04 How it flows



---

## 05 Step by step

---

### 01 Test in the interface first

Open the image tool yourself. Upload a reference ad and a product photo. Ask for the combination. Judge the result with your own eyes.

---

### 02 Check your account tier

Free tiers often refuse multiple reference images, and this approach needs at least two. A paid plan is a prerequisite, not an upgrade.

---

### 03 Stop if the result is bad

If the model cannot do it by hand, no amount of code makes it do it automatically. Walk away and try a different model or the compositing path.

---

### 04 Put the key in the local environment

Open the gear next to the environment selector and add the key as an environment variable. Never paste a key into the chat itself.

---

### 05 Make Claude read the real docs

Tell it to fetch every endpoint, function, and parameter from the official documentation. Its memory of an API is often out of date.

---

### 06 Run exactly one test generation

One live call proves the key works, the endpoint is right, and the parameters are correct. Batching before this just multiplies an error.

---

### 07 Watch for the wrong model version

Nick caught it reaching for the previous version and stopped it on the spot. Read what it is about to call, not just what it produces.

---

08

## Record the working parameters

Ask for the exact parameters it used on the successful test. Those become the fixed settings for everything you build next.

---

---

## 06 Words explained

---

### Feasibility check

A quick manual test to answer one question: can this tool do the job at all? Done before any building.

---

### Endpoint

One specific address in an API you call to make it do one specific thing, like generate an image.

---

### Environment variable

A named value stored outside your code, like an API key. Programs read it without the value ever appearing in the code itself.

---

### Prompt injection

An attack where text on a web page or in a file secretly instructs the model, for example telling it to reveal your keys.

---

### Direct generation

A model that paints every pixel from your description, rather than laying out shapes and text you defined.

---

### Reference image

A picture you hand the model to copy from, like the ad layout you want matched or the product you want featured.

---

---

## 07 Watch out for

---

- Free-tier accounts often cannot upload multiple reference images, and this whole approach needs at least two. Get on a paid plan first.
  - Claude will happily reach for the previous version of the image model. Nick caught it live and stopped it on the spot.
  - Claude cannot read the key back to you, which is deliberate. Do not read that as the key being missing.
  - Skipping the docs step means it calls endpoints from memory, and model memories of APIs go stale fast.
  - Batching before one test passes turns a single wrong parameter into fifty paid failures.
- 

## 08 How it connects

---

### Template Conformity with Bands

The feasibility test tells you the model can combine a reference and a product. Bands are how you then stop it from drifting off that reference.

---

### Compositing vs Direct Generation Models

This is the entry point to the direct generation path, which costs real money and lands about a quarter of the time, so proving it first matters.

---

### Compounding Failure in AI Pipelines

Correcting the wrong model version the second you spot it is the cheapest correction there is, and it is exactly what that framework prescribes.

---

## Creative Pipeline Kickoff

Same starting materials, a reference ad plus product facts, but handed to a pixel model instead of a compositor.

---

---

**Credit.** Framework and approach by Nick Saraev, from the direct image generation section of his Zero to One course on Claude Code for marketing. Written up for the Kape vault by BrewedOps.