

X STOP WATCHING VIDEOS MANUALLY

# Make Claude *watch any video* for you.

A 5-minute setup that turns Claude Code into a video research engine. Hours-long YouTube lectures, Instagram reels, Looms, MP4s — Claude watches the frames, reads the transcript, and hands you the takeaways while you make coffee.

*i.*

## Video is where the *good stuff* hides.

The best content on the internet isn't blog posts anymore. It's podcasts, lectures, walkthroughs, reels — and almost none of it is indexable, searchable, or skimmable.

Every transcript tool on the market reads the words and ignores the screen. That's *half the information*. When a speaker gestures at a chart, demos a UI, or scribbles on a whiteboard, transcript-only tools quietly miss it — and you don't know what you missed.

### What changes when Claude can actually watch

**45 min**VIDEO, INGESTED IN  
UNDER 2 MINUTES**~\$1**PER RUN, CAPPED AT 100  
FRAMES**1,000+**SUPPORTED SITES — NOT  
JUST YOUTUBE

#### THE PROMISE

Paste a URL. Ask a question. Claude has already watched the whole thing frame-by-frame, lined the visuals up with the transcript, and is ready to answer — before you finish typing.

*ii.*

## A video is just *two things*.

Frames and a transcript. That's it. Once you accept that, you don't need a fancy video model — you just need to hand Claude what it already knows how to read: pictures and text.

### The pipeline

1

#### yt-dlp grabs the video

Think "right-click → save video", but it works on basically the whole internet. Pulls subtitles too, if the source has them. Completely free, battle-tested for over a decade.

---

2

#### FFmpeg cuts frames + extracts audio

Screenshots are taken every few seconds across the video. The audio is pulled into a clean little file ready to be transcribed. Both run locally on your laptop — nothing leaves your machine for this step.

---

3

#### Whisper (Groq) fills the gap

If the video had no captions — raw MP4, Loom, Instagram reel — Whisper transcribes the audio. Groq's free tier gives you 2 hours of transcription per hour. Most days you'll never leave the free tier.

---

4

#### Claude reads the flipbook

Frames + per-second timestamped transcript arrive together. Claude flips through the screenshots like a flipbook, reads the script alongside, and knows exactly what's on screen when something is said.

---

iii.

## Set it up in *under 5 minutes.*

You'll need Claude Code (the CLI), Homebrew on macOS, and a Groq account for free transcription. That's all.

### Step 1 — Install Claude Code (if you haven't)

```
# macOS / Linux
$ curl -fsSL claude.ai/install.sh | bash
```

### Step 2 — Drop the watch skill into ~/.claude/skills

```
# Clone the open-source skill (free, MIT)
$ mkdir -p ~/.claude/skills
$ cd ~/.claude/skills
$ git clone https://github.com/bradybonus/claude-watch-skill watch
```

### Step 3 — Run the setup script once

```
# Installs ffmpeg + yt-dlp via brew, scaffolds the .env file
$ python3 ~/.claude/skills/watch/scripts/setup.py
```

### Step 4 — Get a free Groq API key

Go to `console.groq.com/keys`, sign up, create a key. Paste it into `~/.config/watch/.env` as `GROQ_API_KEY=...`. Groq is free, fast, and covers more than you'll ever realistically use.

#### THAT'S IT — TRY IT

Open Claude Code, type `/watch <url>` and paste any YouTube link. Claude will download, extract frames, pull captions, and start answering questions about the video.

*iv.*

## Three ways to put it to *real work*.

Once setup is done, the hardest part is choosing what to point it at. Here are the three that have changed how I consume content.

### CONTENT RESEARCH

#### Decode any viral video's hook

Paste a winning reel. Ask: "Break down the hook — visual setup, exact words, where the pattern interrupt lands, and what's on screen at the hook moment." Stuff that used to take 10 minutes of pausing and scrubbing is now a single paste.

### BUG HUNTING

#### Debug from a screen recording

Record a 30-second screen capture of a UI bug. Drop it into Claude. Ask: "What happens right before the crash?" Claude reads the frames around that moment, finds the state change, and tells you the exact frame the issue starts. Saves hours per bug.

### SECOND BRAIN

#### Auto-feed a knowledge base

Hand Claude a list of creators you respect. It watches each new video — frames, audio, the lot — comes back with structured notes on what made the video work, and files them straight into Obsidian or Notion. Your library compounds while you sleep.

### Pro tip — zoom into a single moment

On long videos, add `--start 2:15 --end 2:45` to focus the frame extraction on a 30-second window. You get denser sampling without burning the whole context budget.

v.

## "But won't this *torch my tokens?*"

It doesn't. The skill caps any video at 100 frames, so a 30-minute video and a 1-hour video cost basically the same. Here's the rough math.

~\$1

PER VIDEO RUN (CLAUDE USAGE)

\$0

YOUTUBE CAPTIONS — PULLED FREE

\$0

GROQ WHISPER — FREE TIER COVERS IT

### Real-world stress test

Brad Bonus, who built the skill, ran every test in his demo three times in parallel — over **5 hours of video** watched live by Claude with transcription — and burned less than **10% of a single Claude Code session**. Two weeks of daily use later, he was still on Groq's free tier.

### Where it gets expensive

Only one thing scales the cost: *resolution*. Frames default to 512 px wide. Bumping to 1024 px roughly quadruples the image-token cost per frame. Only do it when you need to read on-screen text (slide decks, code, fine print).

#### RULE OF THUMB

If a video is under 10 minutes, let the skill run full-mode. If it's over 30 minutes, use `--start` and `--end` to focus on the section you care about. You'll get better answers *and* spend fewer tokens.

*vi.*

# You'll never watch a video the *old way* again.

The first time Claude crunches a 45-minute lecture in two minutes and answers your question with timestamps, you'll feel the shift. From here on, videos become source material, not a time sink.

## YOUR 5-MINUTE CHECKLIST

- ✓ Install Claude Code on your machine
- ✓ Clone the watch skill into `~/.claude/skills/watch`
- ✓ Run the setup script — auto-installs ffmpeg & yt-dlp
- ✓ Grab a free Groq API key, paste it into the `.env`
- ✓ Type `/watch <url>` and watch Claude work

Get more guides at [skillies.ai](https://skillies.ai) →