

Air-O-Clean — How It Works

Can a photo of the sky estimate air pollution, and how would you know if it could? An iPhone app and a website that test that question on every scan.

The science

Fine particles in the air (PM2.5) scatter sunlight. That has three visible effects: distant detail loses contrast, the sky's blue washes out towards white, and the horizon blurs. The more particles, the stronger each effect, which is why visibility falls as pollution rises. Gases are a different story: nitrogen dioxide can tint the sky faintly brown, sulphur dioxide even less, and carbon monoxide and ozone are invisible. So I expect a photo to track particulates reasonably well and gases poorly. Finding where the method breaks is part of the result.

How a scan works

- 1 **Capture.** Three frames are taken and combined pixel by pixel using the median, which throws out a single bad frame (a bird, a glare flash) instead of blending it in.
- 2 **Observe.** Google's Gemini vision model rates the photo on fixed 0–1 scales: haze, sky blueness, horizon sharpness, colour tint, cloud cover, likely fog. It is **forbidden to judge pollution**; it only describes what it sees.
- 3 **Judge.** My scoring model does the judging, transparently (see below).
- 4 **Check.** The app fetches the nearest government sensor's reading (OpenAQ) and stores the prediction and the truth side by side, without the photo. Every capture is therefore an experiment: a prediction and its answer, one row in a growing dataset.

The mathematics, step by step

The scoring model is deliberately transparent — every number in it is one I chose and can defend.

Step 1 — Turn the photo into numbers. The vision model reports three visual measurements, each on a 0-to-1 scale: haze (how milky the air looks), blueness (how saturated the sky's color is), and sharpness (how crisp the horizon line is).

Step 2 — Combine them into a single haze score. Pollution makes haze go *up* while blueness and sharpness go *down*, so the score uses each variable in the direction that means "worse air," weighted by how strongly it signals particulates:

$$\text{haze score} = 100 \times [0.5 \cdot \text{haze} + 0.3 \cdot (1 - \text{blueness}) + 0.2 \cdot (1 - \text{sharpness})]$$

Haze gets half the weight because it is the most direct optical effect of particulate pollution: particles scatter light, washing out contrast. A crisp blue sky scores near 0 and a sky merged into grey murk scores near 100.

Step 3 — Convert the score to a rating. The 0–100 scale is divided into the six standard EPA air-quality bands (Good below 15 ... Hazardous above 90), so the app's output can be compared directly against official sensor categories.

Step 4 — Recalibrate. The weights (0.5 / 0.3 / 0.2) and band boundaries are my hand-set starting hypothesis. Once enough photo–sensor pairs accumulate, I re-fit them from the data and report whether the calibrated model beats the hand-tuned one — a genuine before-and-after experiment on my own assumptions.

How I test it

A pair only counts if the sensor is within 10 km and its reading within 60 minutes of the photo. I measure **Spearman rank correlation** (does a worse-looking sky mean worse air?), a **confusion matrix** (where predictions go wrong, not just how often), and exact and within-one-band accuracy. Fog is analysed separately, because it looks like pollution but isn't. The target is about 100 validated pairs before refitting.

Phone app vs website

Both run the same scoring model, kept identical by a shared settings file and 12 shared test cases, and both feed one dataset. The important difference is camera control.

	iPhone app	Website
Getting it	Installed app	Any phone browser, no install
Camera exposure and white balance	Always locked before capture	Locked only on some Android phones; never on iPhone , where every browser uses Safari's engine
Recorded as	ios_locked	web_locked or web_auto
Three frames	Three separate photos	Three frames from the live camera, about 0.3 s apart
Camera settings logged	ISO and exposure time, every scan	Only where the browser reports them
Talks to Gemini	Directly	Through the website's own server, so the access key never reaches the browser
Scan history	Kept on the phone	Kept in the browser; Safari erases it after 7 days without a visit unless the site is added to the Home Screen

Why the difference matters. A camera left on automatic adjusts itself to each scene, which can make a hazy sky and a clear sky look more alike than they are, flattening the very signal being measured. Rather than hide this, every scan records how it was taken, and the analysis compares accuracy for each mode. That turns a weakness of the website into a second question: *how much does locking the camera actually improve accuracy?*

Privacy

Live and collecting data from friends and family in India and the US. Photos are never stored or shared by the project (they go to Google's Gemini for analysis only), locations are rounded to about 110 m, and contributors can opt out.

Why it's honest

I am not a programmer. The code was written with Claude Code, an AI coding tool, and I credit that openly. What I own is everything this document describes: the question, the scoring model, the validation methodology, and the interpretation of the results. I can defend every decision here without explaining a line of code — and I think knowing *what to build and how to prove whether it works* is exactly the skill an economics and mathematics education trains.