cv2.cvtColor(images\[i], cv2.COLOR_RGB2GRAY) - *Grayscale images have only one channel, representing brightness. Instead of each pixel having red, green, and blue values, each pixel has a single intensity value, typically ranging from 0 (black) to 255 (white).*

cv2.equalizeHist(gray) - takes a grayscale image (gray) and redistributes its pixel intensity values to enhance the contrast across the image.

cv2.cvtColor(images\[i], cv2.COLOR_RGB2LAB) - ***LAB** is a perceptually uniform color space designed to approximate human vision more closely than RGB.*

It has three channels:

L – Lightness (brightness), from 0 (black) to 100 (white)

***A** – Green to Red component*

B – Blue to Yellow component

cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))

Divides the image into small tiles (defined by `tileGridSize`)

***Applies histogram equalization** in each tile separately*

Then **blends** the results to avoid artificial borders

***Limits contrast amplification** (via `clipLimit`) to prevent noise from getting boosted*

So by converting to LAB, we can:

1. Isolate the **lightness** component (`L`)
2. Apply **CLAHE only to L** (to enhance contrast)
3. Keep color information (A & B) untouched
4. Recombine L, A, B and convert back to RGB

This preserves the **original colors** while **boosting contrast**, without introducing weird color artifacts.

* cv2.GaussianBlur(noisy, (15, 15), 0) - Gaussian blur uses a **Gaussian function** (bell-shaped curve) to **weight surrounding pixels** when computing the value of each output pixel. This creates a natural, smooth blur — much like what you see when you slightly defocus a camera.\
\
Copied from: ChatGPT - <https://chatgpt.com/>