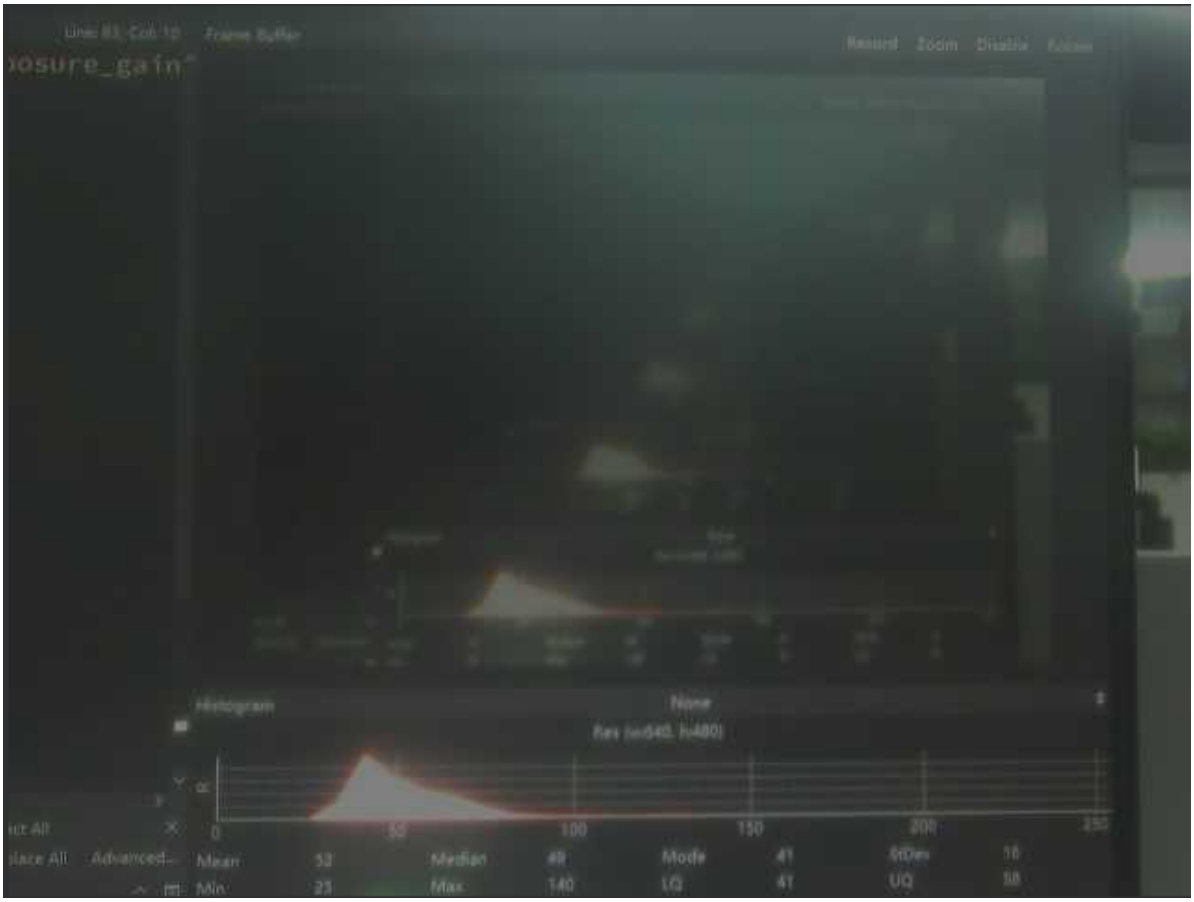
Exposure Adjustment

Example Results

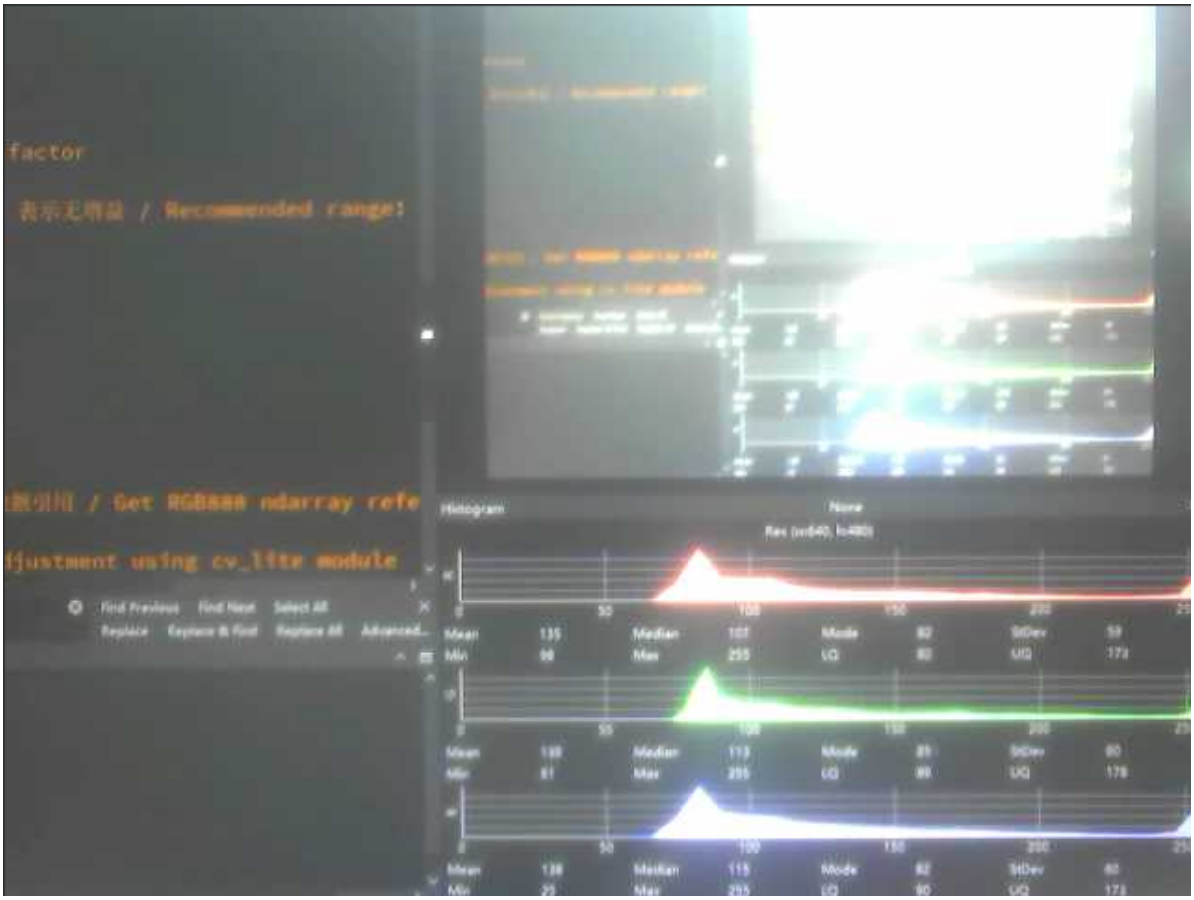
Run this section's example code [Source Code/06.cv_lite/15.rgb888_adjust_exposure.py]

In this section, we will use the `cv_lite` extension module to implement exposure gain adjustment in the RGB888 format on an embedded device.

> The exposure gain factor is set to 0.5



> The exposure gain factor is set to 1.5



Code Overview

Importing modules

```
import time, os, sys, gc
from machine import Pin
from media.sensor import * # Camera interface
from media.display import * # Display interface
from media.media import * # Media manager
import _thread
import cv_lite # AI CV extension module (including exposure adjustment
interface) / AI CV extension with exposure support
import ulab.numpy as np # ulab array library (for image data processing) /
NumPy-like ndarray for MicroPython
```

Setting image size

```
image_shape = [480, 640] # Height x Width
```

Define the image resolution to 480x640 (height x width). The camera and display module will be initialized based on this size later.

Initialize the camera (RGB888 format)

```
sensor = Sensor(id=2, width=1280, height=960, fps=90)
sensor.reset()
sensor.set_framesize(width=image_shape[1], height=image_shape[0])
sensor.set_pixformat(Sensor.RGB888) # Set pixel format to RGB888
```

- Initialize the camera, set the resolution to 1280x960 and the frame rate to 90 FPS.
- Resize the output frame to 640x480 and set it to RGB888 format (three-channel color image, suitable for processing exposure adjustment of color images).

Initialize the display module

```
Display.init(Display.VIRT, width=image_shape[1], height=image_shape[0],
to_ide=True, quality=50)
```

Initialize the display module, use the virtual display mode (`Display.VIRT`), and the resolution is consistent with the image size. `to_ide=True` means that the image will be transferred to the IDE for virtual display at the same time, and `quality=50` sets the image transfer quality.

> This will not be displayed on the K230 screen. If you want it to be displayed on the K230 screen, please change the first parameter to Display.ST7701

Initialize the media manager and start the camera

```
MediaManager.init()
sensor.run()
```

Initialize the media resource manager and start the camera to capture the image stream.

Set the sensor analog gain (optional)

```
gain = k_sensor_gain()
gain.gain[0] = 20 # Set gain for channel 0
sensor.again(gain) # Apply analog gain
```

Set the analog gain parameters of the camera, adjust the brightness and contrast to optimize the image quality, `gain[0] = 20` is used to increase the image brightness.

Set the exposure gain parameter

```
exposure_gain = 0.5 # Recommended range: 0.2 to 3.0; 1.0 means no gain
clock = time.clock() # Start FPS timer
```

- `exposure_gain`: Exposure gain factor, used to adjust image brightness. A value less than 1.0 reduces brightness, a value greater than 1.0 increases brightness, and 1.0 means no gain.
- `clock`: Used to calculate frame rate.

Image processing and exposure adjustment

```
while True:
    clock.tick() # Start timing

    # Capture a frame
    img = sensor.snapshot()
    img_np = img.to_numpy_ref() # Get RGB888 ndarray reference (HWC)

    # Apply exposure adjustment using cv_lite module
    exposed_np = cv_lite.rgb888_adjust_exposure_fast(
        image_shape,
        img_np,
```

```
        exposure_gain
    )

    # Wrap processed image for display
    img_out = image.Image(image_shape[1], image_shape[0], image.RGB888,
                          alloc=image.ALLOC_REF, data=exposed_np)

    # Display image
    Display.show_image(img_out)

    # Cleanup memory and display FPS
    gc.collect()
    print("adjust exposure:", clock.fps())
```

- **Image capture**: Acquire an image frame using `sensor.snapshot()` and convert it to a NumPy array reference using `to_numpy_ref()`.
- **Exposure adjustment**: Call `cv_lite.rgb888_adjust_exposure_fast()` to adjust the exposure gain, returning the adjusted image as a NumPy array.
- **Image packaging and display**: Package the processed result into an RGB888 image object and display it on the screen or in an IDE virtual window.
- **Memory management and frame rate output**: Call `gc.collect()` to clean up memory and print the current frame rate using `clock.fps()`.

Resource release

```
sensor.stop()
Display.deinit()
os.exitpoint(os.EXITPOINT_ENABLE_SLEEP)
time.sleep_ms(100)
MediaManager.deinit()
```

Parameter adjustment instructions

- `exposure_gain`: Exposure gain factor. A value less than 1.0 will reduce the image brightness, while a value greater than 1.0 will increase the image brightness. It is recommended to adjust it within the range of 0.2 to 3.0, and 1.0 means no gain.
- `gain.gain[0]` : Analog gain value, used to adjust the initial brightness of the camera sensor. Adjust according to the ambient light conditions. It is recommended to test from 10 to 30 to optimize image quality.