آزمایش دوم، تصحیح جریان تاریک:

بردیا عالیا

ن، بردیا حسن‌پور

۴۰۱۱۰۰۸۷۴، ۴۰۱۱۰۰۷۱۱

دوربین مورد استفاده: Nikon D90

ISO: 200

۱. نحوه انجام آزمایش:

ابتدا دوربین را روشن می‌کنیم و حساسیت آن را روی عدد ۲۰۰ تنظیم می‌کنیم. در دوربین را بسته نگه میداریم و با هر سرعت شاتر 1/4000, 1/1000, 1/100, 1/10, 1, 10, 30 Sec ده عکس میگیریم. باید توجه کنید که بین هر عکس چند ثانیه به دوربین زمان دهید تا دوربین خنک شود.

داده ها را به کامپیوتر میفرستیم.

۲. پردازش عکس ها:

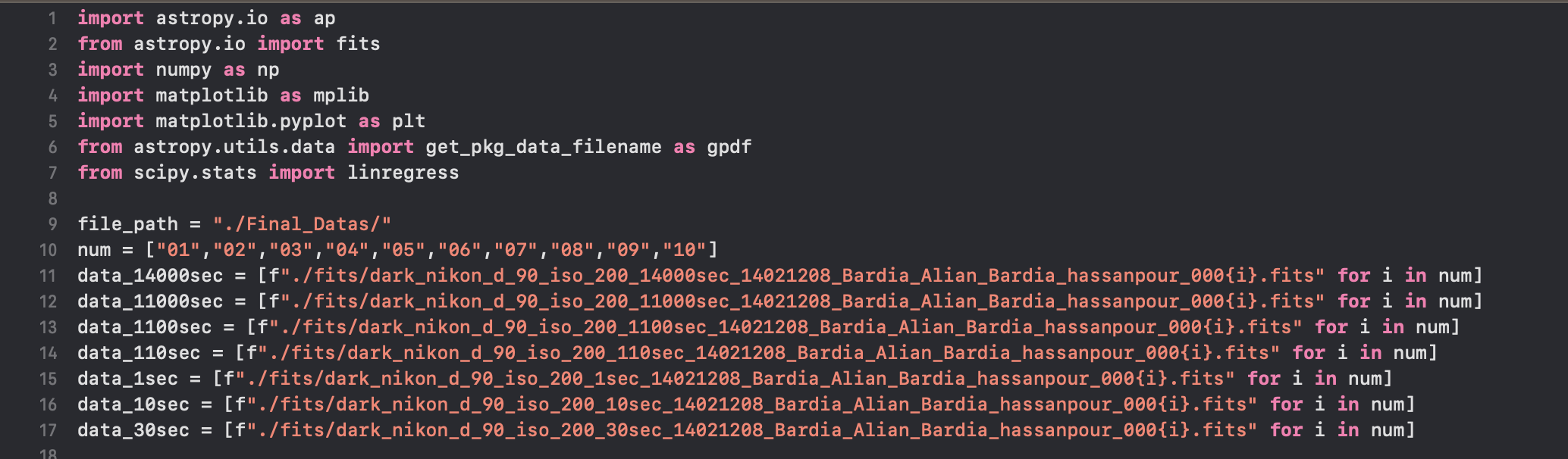
۲.۱. تبدیل عکس ها به فایل .fits

فرمت عکس های دوربین Nikon D90 به صورت .NEF می‌باشد. ابتدا باید عکس ها را به فرمت .fits تبدیل کنیم. فرمت مورد نظر می‌تواند پیچیدگی های زیادی داشته باشد اما برای این آزمایش با توجه به ساده بودن داده ها و به دلیل تبدیل ساده برنامه استفاده برای تبدیل عکس ها به .fits این پیچیدگی مساله پیچیده ای برای آزمایش ما نیست.

برای تبدیل عکس ها به فرمت .fits از برنامه Siril استفاده می‌کنیم. برای فرمت مورد نظر و تبدیل عکس ها به فرمت مورد نظر در .fits در تنظیمات به بخش FITS Options می‌رویم و در بخش FITS Formats -> Extension فرمت فایل دریافتی را .fits می‌کنیم سپس در بخش FITS/SER Debayer -> Debayer -> Bayer/mosaic pattern گزینه RGGB را انتخاب میکنیم. سپس در برنامه در بخش Conversion فایل عکس ها را داده و با انتخاب گزینه Debayer و Symbolic link عکس هارا به فرمت .fits تبدیل می‌کنیم. عکس تبدیل شده دارای تنها یک HDU می‌باشد که در آن سه مولفه رنگی تصاویر ذخیره اند.

۲.۲. برنامه پردازش عکس تاریک:

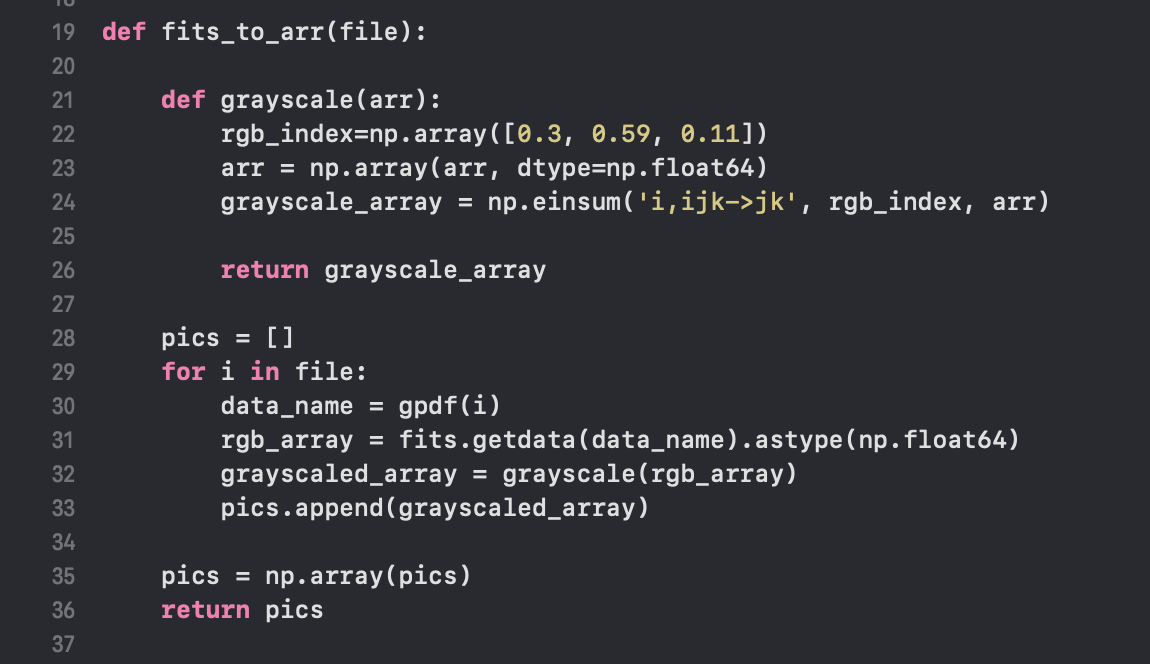
ابتدا آدرس فایل ها را برای هر سرعت شاتر در یک لیست به صورت زیر میریزیم:



file\_path آدرس مکانی است که داده های نهایی در آن ذخیره خواهند شد.

۲.۲.۱. تبدیل فایل .fits به آرایه پایتون به صورت سیاه سفید:

در این بخش تابع زیر را تعریف میکنیم:



ابتدا داخل تابع تابع دیگری برای سیاه و سفید کردن تصویر با ضرایب مربوط به هر فیلتر رنگی کرده ایم که با einsum آرایه یک بعدی ضرایب را در آرایه سه بعدی آرایه فایلمان به صورتی ضرب می‌کند که ضرایب در شمارش هر پیکسل در هر فیلتر رنگی ضرب شوند.

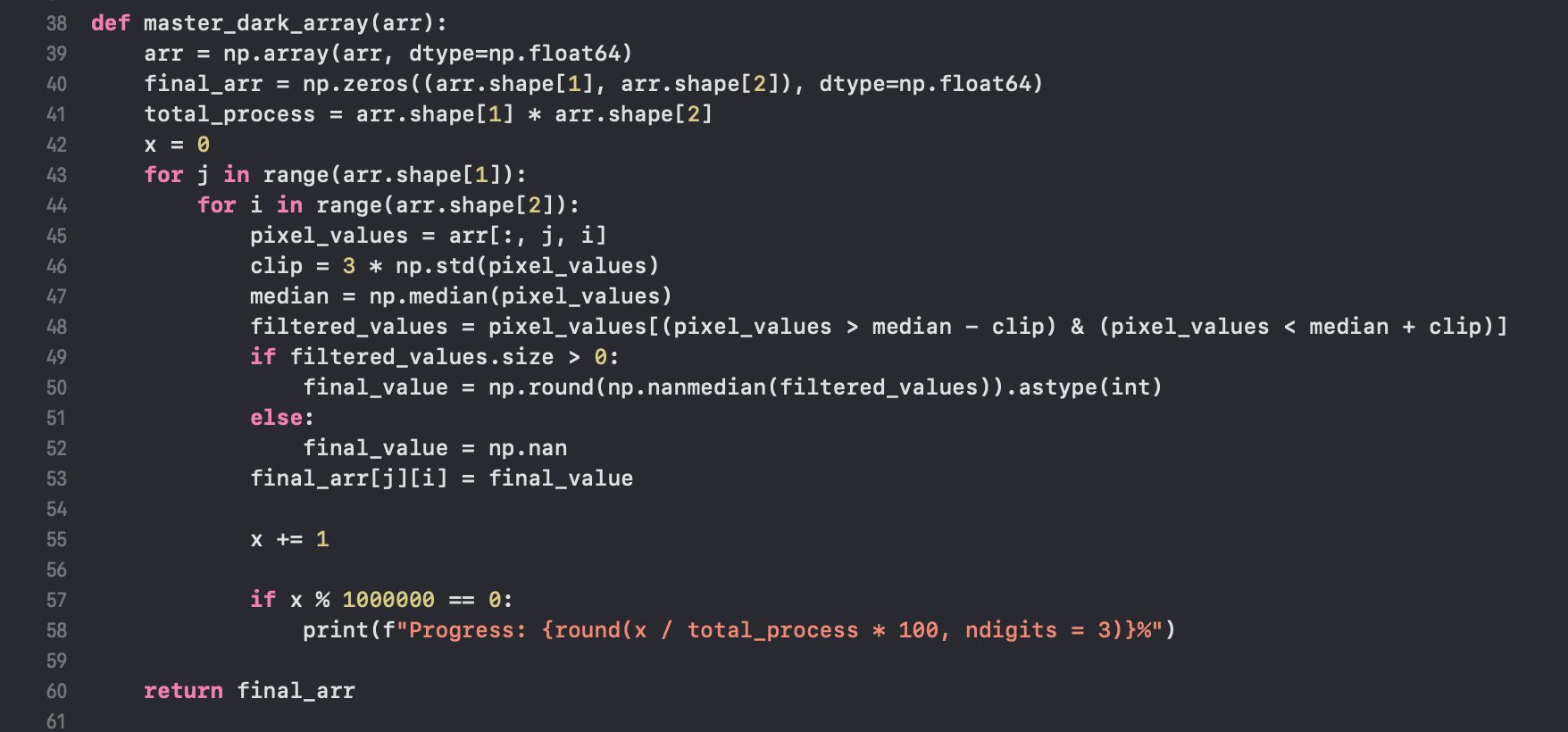
در قدم بعدی برای هر عکس در لیست آدرس فایلمان با get\_package\_data\_filename به یک متغیر تبدیل و سپس با fits.getdata آرایه عکس ها در سه فیلتر RGB را می‌گیریم. نکته: این روش تنها برای فایل های فیتز ساده که تنها یک HDU دارند قابل انجام است و روش کلی برای تبدیل کردن فایل فیتز به آرایه یک چیز دیگر است. اما به این دلیل که این روش برای فایل های ساده فیتز سریع تر است از این روش استفاده می‌کنیم.

در قدم بعد آرایه سه بعدی عکس هایمان در فیلتر RGB را با تابع grayscale به یک عکس سیاه سفید که یک آرایه دو بعدی است تبدیل و آن را به یک لیست اضافه میکنیم.

در نهایت لیست را آرایه کرده و آن را برمی‌گردانیم. آرایه برگردانده شده یک آرایه سه بعدی شامل ده عکس است.

۲.۲.۲. ترکیب عکس ها و Sigma clipping:

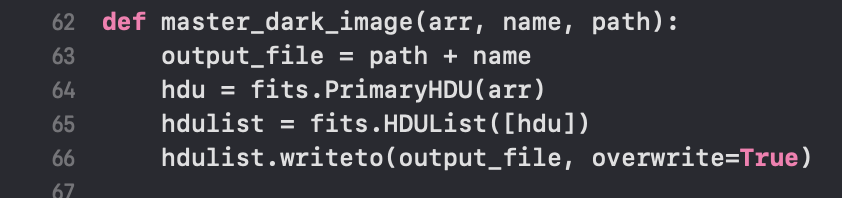
حال که عکس هایمان را به یک سری آرایه دو بعدی تبدیل کردیم که در آن Count معادل سیاه سفید پیکسل ها است، هر ده عکس با یک زمان نوردهی را ترکیب کرده و آن را Sigma clip می‌کنیم.



ابتدا تابع master\_dark\_array را تعریف می‌کنیم که قرار است آرایه عکس تاریک نهایی را به ما بدهد. به دلیل آنکه آرایه غیرقابل تغییر در اندازه است ابتدا آرایه عکس نهاییمان را با ابعاد عکس‌هایمان می‌سازیم و آن را پر از ۰ می‌کنیم. برای دانستن میزان پردازش‌های گذشته نیز تعداد تمام پیکسل‌های عکسمان را در ‌total\_process ذخیره میکنیم تا در ادامه تابع درصد پروسس شدن عکس نهایی را خروجی بگیریم.

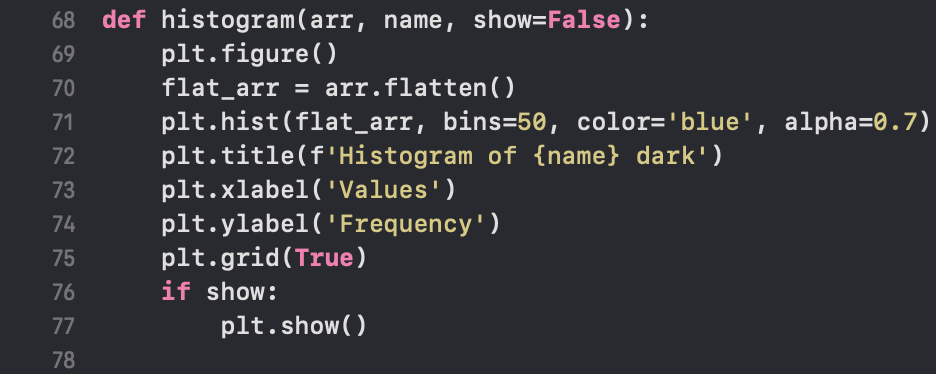
سپس با دو for بروی پیکسل j-i ام هر ده عکس میرویم و روی آن Sigma clip بروی میانه با دامنه سه برابر انحراف معیار انجام می‌دهیم و داده های پرت را به NaN تبدیل می‌کنیم. در نهایت با nanmedian میانه داده های فیلتر شده را می‌گیریم و به عنوان داده پیکسل معادل در سطر j و ستون i عکس نهایی می‌ریزیم. در نهایت آرایه عکس معادل را از تابع برمی‌گردانیم.

۲.۲.۳. تبدیل آرایه نهایی به فایل .fits:



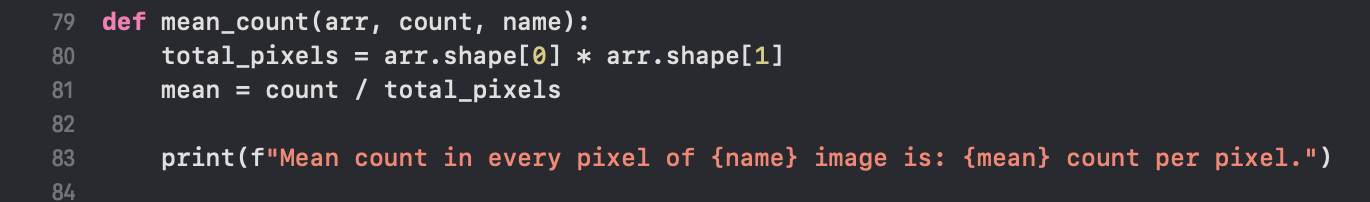
تابع بالا با قابلیت های کتابخانه astropy.io.fits آرایه عکس نهایی را به فایل فیتز تبدیل کرده و ده آدرس و اسم مورد نظر ذخیره می‌کنید. اگر فایل با همان اسم موجود باشد این تابع آن را بازنویسی می‌کند.

۲.۲.۴. ساخت هیستوگرام:



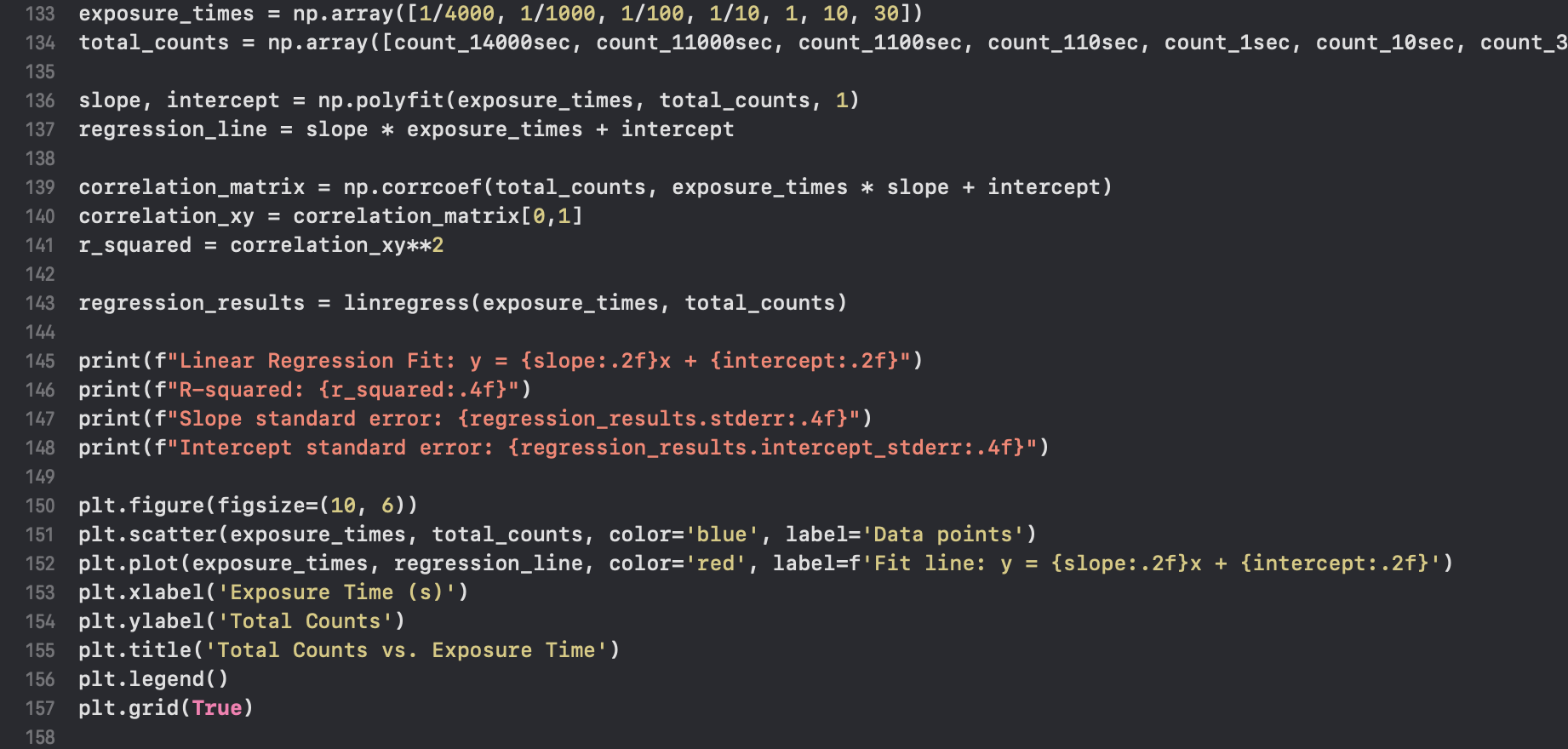
تابل بالا یک هیستوگرام تعداد پیکسل ها بر حسب Count اشان را می‌سازد اما ما دستور نشان دادن این نمودارها را در نهایت می‌نویسیم تا همه نمودار هایمان رسم شوند.

۲.۲.۵. میانگین Count در هر پیکسل:



تابع بالا میانگین Count در هر پیکسل را گزارش می‌دهد.

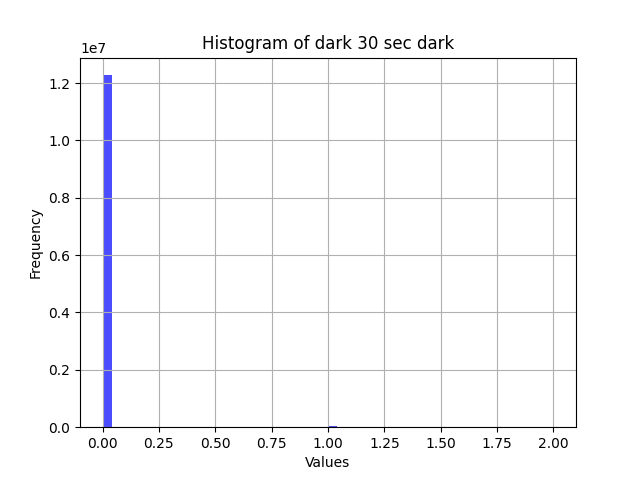
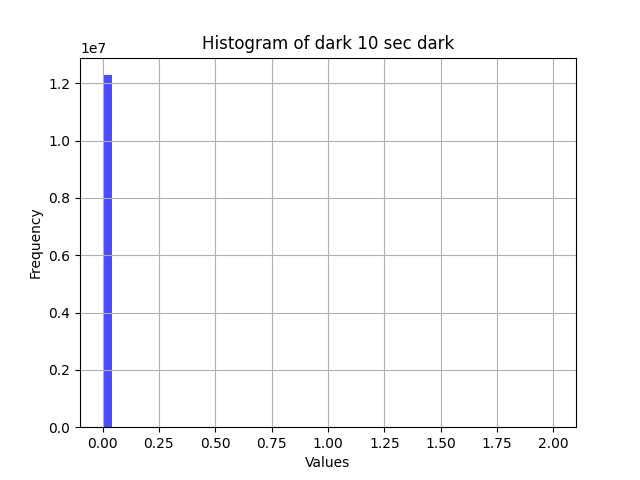
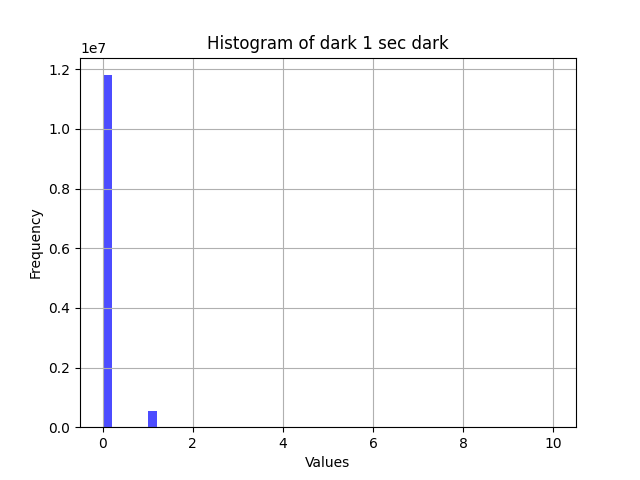
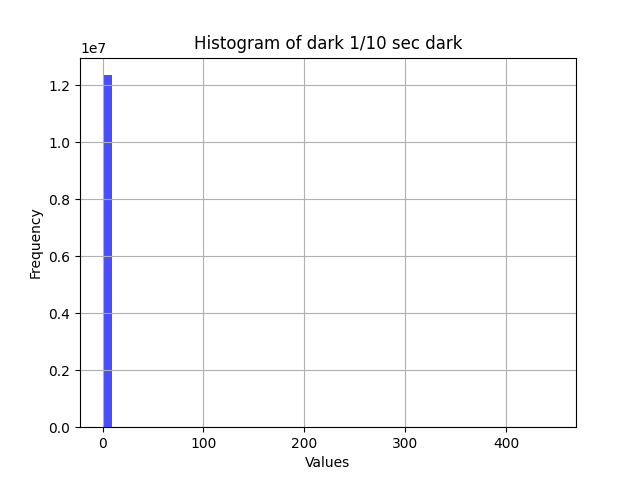
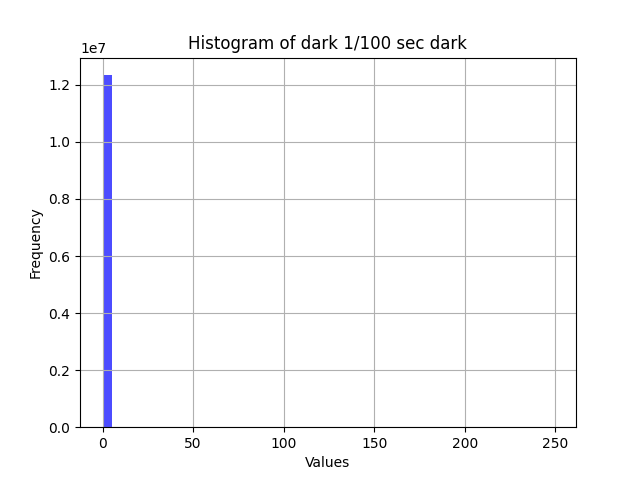
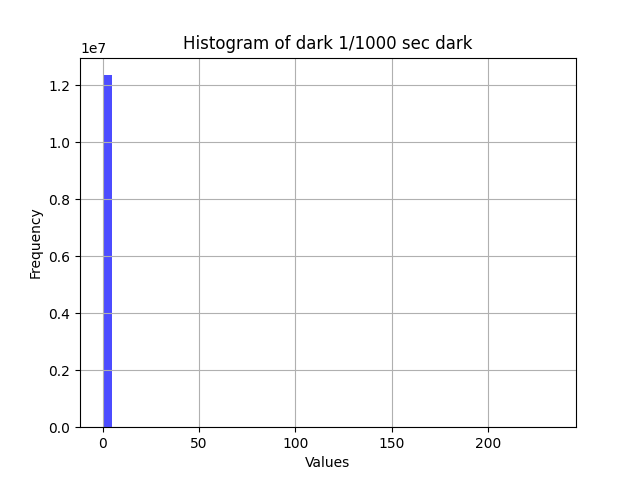
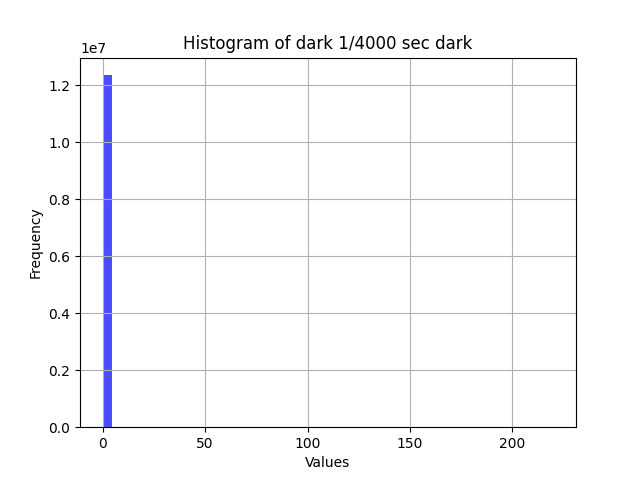
۲.۲.۶. رسم نمودار و برازش خط:



کد بالا نمودار تعداد Count تصویر بر حسب زمان را رسم میکنید و به آن یک خط برازش می‌کند.

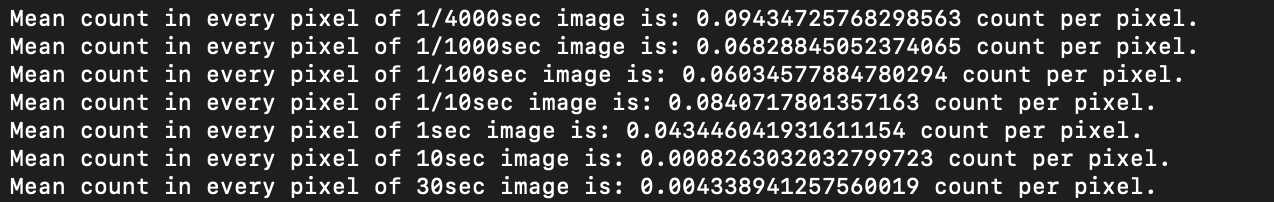
۳. خواسته ها:

۳.۱. هیستوگرام:



۳.۲. میانگین Count:

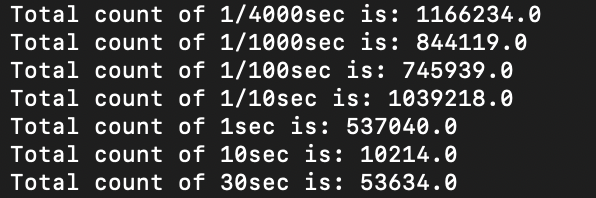
با توجه به برنامه میانگین ها به صورت زیر است:



یک نتیجه گیری حدودی می‌تواند این باشد که نورگیری بیشتر باعث کمتر شدن نویز می‌شود.

۳.۳. انرژی دریافتی هر مدت زمان:

با توجه به داده برنامه Count های کل به صورت زیر می‌باشد:



اگر به طور میانگین انرژی هر فوتون Count را برابر انرژی فوتون 550nm بگیریم:

‌

پس برای شدت داریم:

که A مساحت سطح سنسور دوربین میباشد.().

نمودار شدت زمان به صورت زیر است:

|  |  |
| --- | --- |
| I(w/m^2) | T (sec) |
| 4.51\*10^-3 | 1/4000 |
| 8.17\*10^-4 | 1/1000 |
| 7.22\*10^-5 | 1/100 |
| 1.01\*10^-5 | 1/10 |
| 5.20\*10^-7 | 1 |
| 9.89\*10^-10 | 10 |
| 1.73\*10^-9 | 30 |

*نمودار شدت زمان به صورت زیر است:*