آزمایش تابع پخش نقطه ای (PSF):

۱. پردازش داده ها:

۱.۱.star\_identifyer.py :

در این برنامه ستاره های داخل ۳ عکس را پیدا می‌کنیم و آن را به صورت یک سری عکس با فرمت .fits که ابعاد آن ها 100x100 است ذخیره می‌کنیم. این برنامه اینگونه کار می‌کند که ابتدا عکس ها را باز می‌کنیم و آن ها را به fits تبدیل می‌کنیم. سپس با scipy.gaussian\_filter عکس ها را به یک عکس با قله های گاوسی تبدیل می‌کنیم. در قدم بعدی محدوده هایی که بالای ۱۰۰۰ مقدار دارند را محدوده های پتانسیل دار برای ستاره بودن انتخاب می‌کنیم. در قدم بعدی با binary\_erosion, binary\_dialation نویز ها را کم می‌کنیم. سپس مرکز هر کدام از نواحی پتانسیل ستاره بودن را با مرکز جرم پیدا می‌کنیم و بعد برای هرستاره یک ناحیه 101x101 که مرکز آن مرکز ستاره مرکز آن است را جدا می‌کنیم و در نهایت این نواحی را به صورت یک فایل fits در می‌آوریم و خروجی می‌گیریم به طوری که هدر اول آن عکس و هدر دوم آن مختصات مرکز ستاره بروی عکس اصلی است.

۱.۲. PSF.py:

ابتدا دستی قبل از برنامه عکس هایی که از برنامه قبلی گرفتیم را چک می‌کنیم و عکس های دارای دو ستاره و عکس های خراب را حذف می‌کنیم. برنامه پیدا کردن ستاره به دلیل هات پیکسل های زیاد یک سری عکس‌ (در تصویر دو هات پیکسل ها بروی سحابی برای برنامه مانند ستاره بودند) خراب به ما خروجی می‌دهد.

شماره عکس های درست را به برنامه می‌دهیم. در مرحله بعد یک حد برای پیکسل های اشباع شده تایین می‌کنیم. این حد را اینگونه پیدا می‌کنیم که یک بار برنامه را بدون لیمیت اجرا می‌کنیم و مشاهده می‌کنیم که ستاره هایی که دارای پیکسل با مقدار بالای حدود ۳۰۰۰ هستند پخ هستند که به معنای اشباع است پس حد ما برابر ۳۰۰۰ خواهد بود. در قدم بعدی یک تابع تعریف می‌کنیم که بفهمید یک محوطه اشباع شده یا یک تک پیکسل. این برای پیدا کردن فرق ستاره های اشباع شده با پیکسل های داغ را در تصویر پیدا می‌کند. سپس ستاره های اشباع شده را از لیست ستاره ها حذف می‌کنیم (ستاره های اشباع شده یک محوطه اشباع شده دارند نه یک تک پیکسل اشباع شده) و موقعیت ستاره های مورد تایید را روی عکس ها نمایش می‌دهیم. در قدم بعدی در هر عکس باقی مانده با سیگما کلیپ هر پیکسل در مربعی ۵ در ۵ حول هر پیکسل، پیکسل های منفرد که همان هات پیکسل هایمان هستند را سیگما کلیپ می‌کنیم. سپس با استفاده از scipy.optimize.curve\_fit تابع گاوسی سه بعدی به عکس هایمان برازش می‌کنیم. تابع گاوسی سه بعدی به صورت زیر می‌باشد:

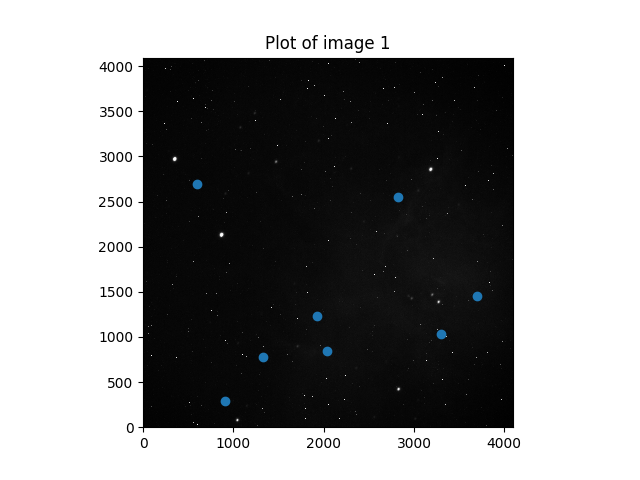
به طوری که:

و در آخر نمودار های مورد نظر را رسم میکنیم و خروجی های مورد نظرمان را می‌گیریم. (تمامی خروجی های نهایی میانه داده ها هستند.

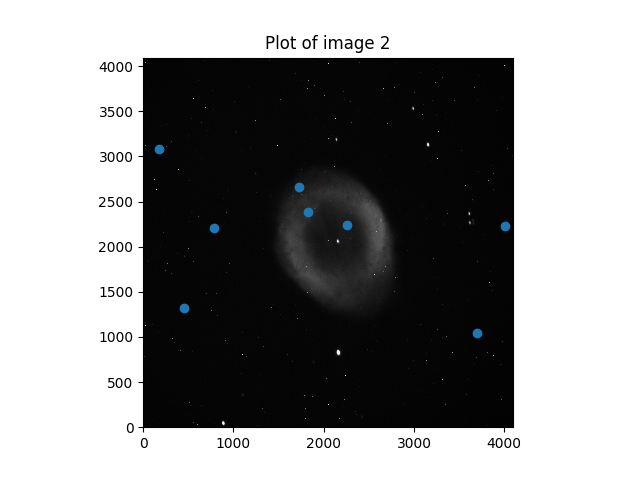
ک نکته در برنامه مان اینست که ما یک لیست کلی از عکس ها می‌سازیم و برای اینکه بفهمیم کدام ستاره برای کدام عکس است ابتدا حلقه ای می‌سازیم و برای هر ستاره یک لیست اندیس می‌سازیم که در آن تنها ابتدا اندیس های ستاره های هر عکسی که با آن کار داریم وجود داشته باشد که آنها را در مرحله اول روی هر عکس مشخص و پلات کنیم. سپس از آنها اندازه می‌گیریم و آن را در یک متغیر ذخیره می‌کنیم که با آن در آینده بدانیم که کدام ستاره در لیست برای کدام عکس است (ستاره های عکس یک در ابتدای لیست و بعد آن ستاره های عکس دو و بعد آن ستاره های عکس سه در لیست ستارگان ذخیره‌اند).

۲. خواسته ها:

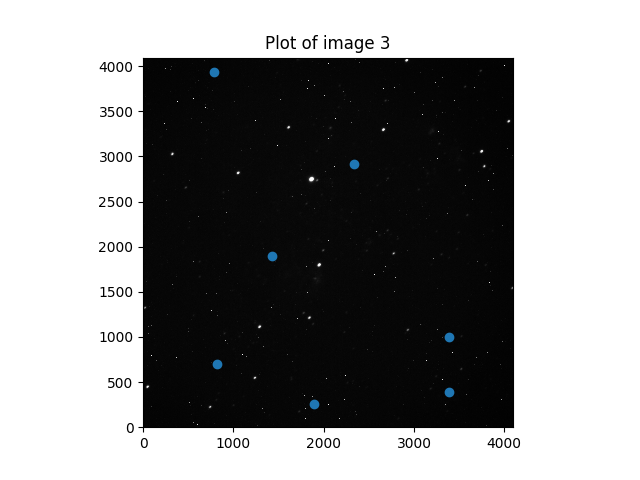
مکان ستاره های انتخاب شده در هر عکس به صورت زیر است:



شکل ۱. موقعیت ستارگان بروی عکس یک

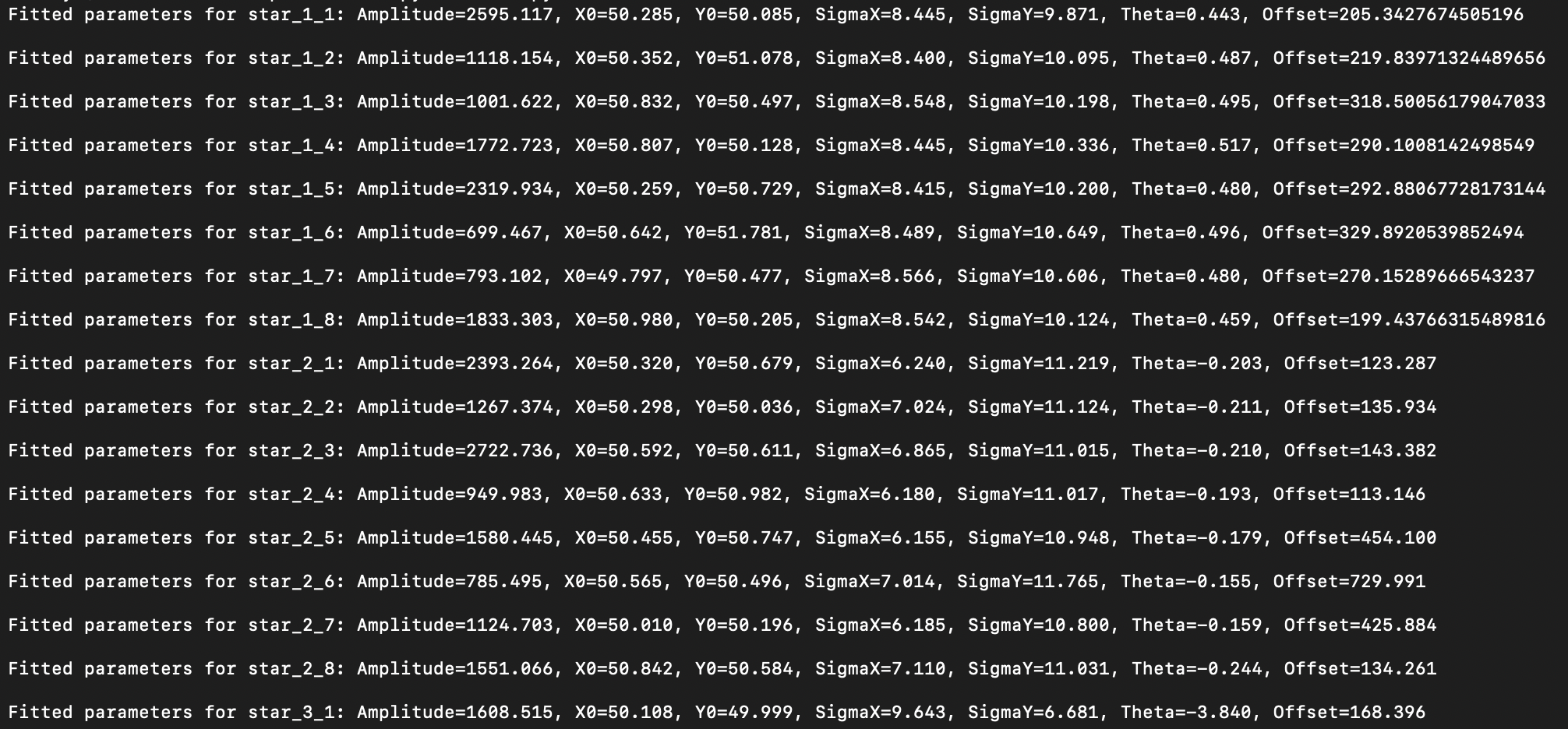


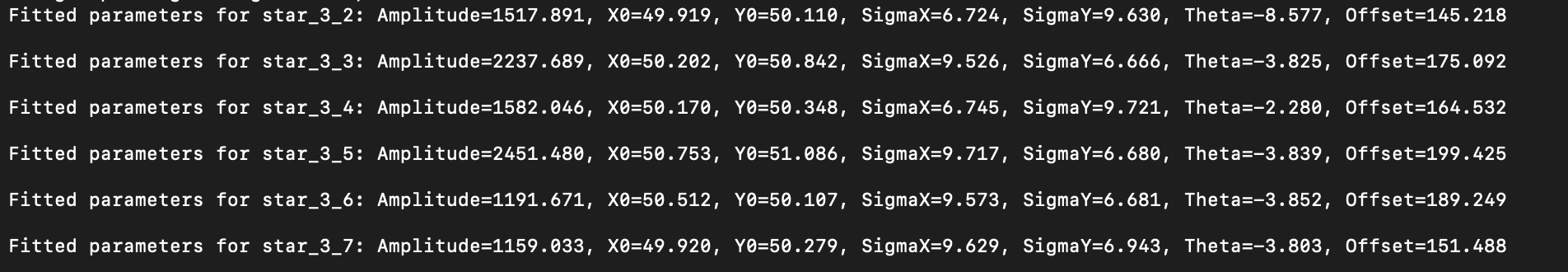
شکل ۲. موقعیت ستارگان بروی عکس دو



شکل ۳. موقعیت ستارگان بروی عکس سه

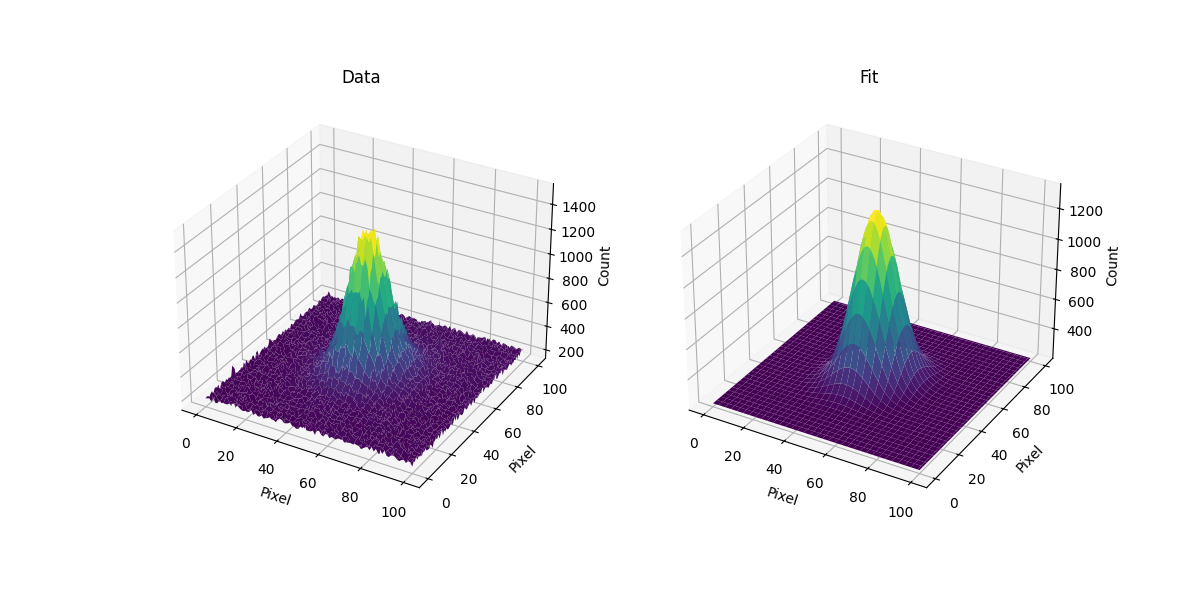
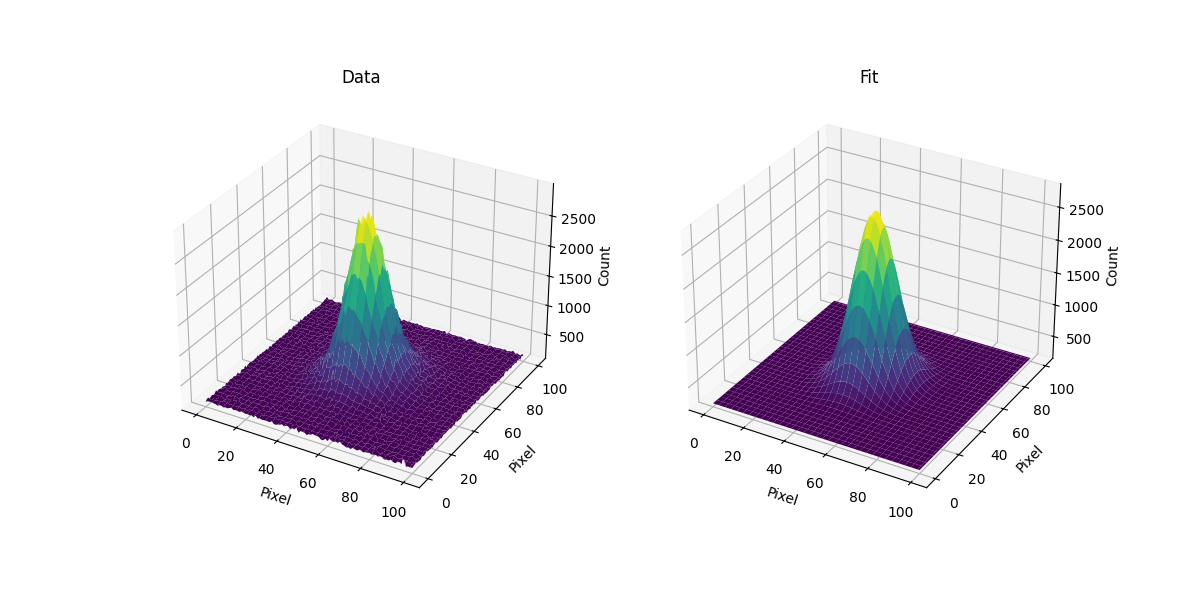
از طرفی داده های فیت گاوسی ستاره های سه عکس به شکل زیر است:





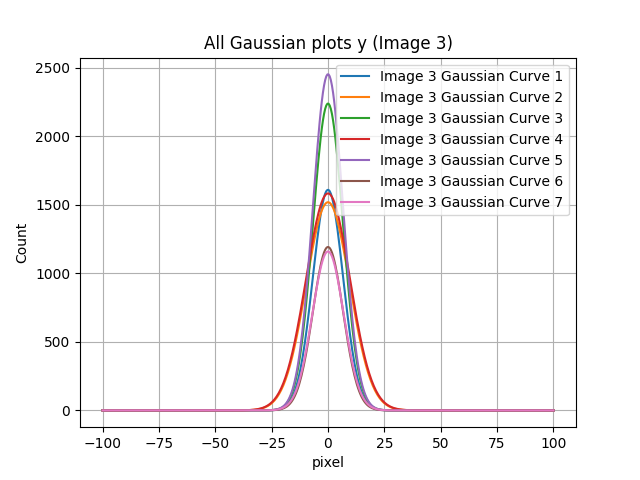
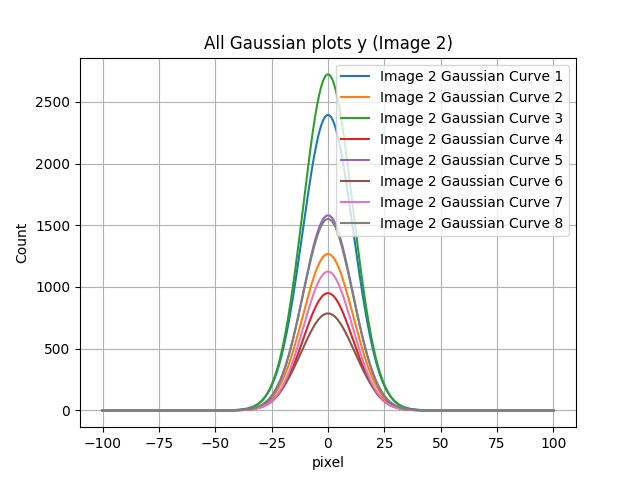
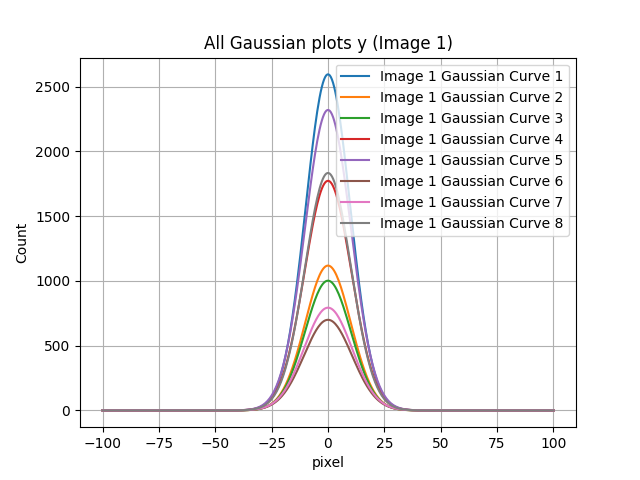
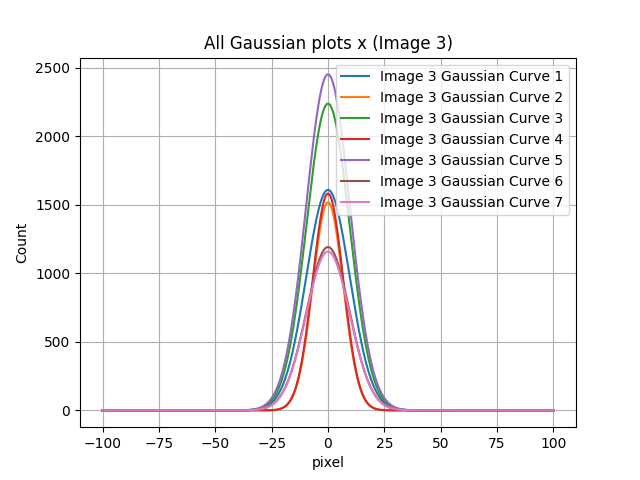
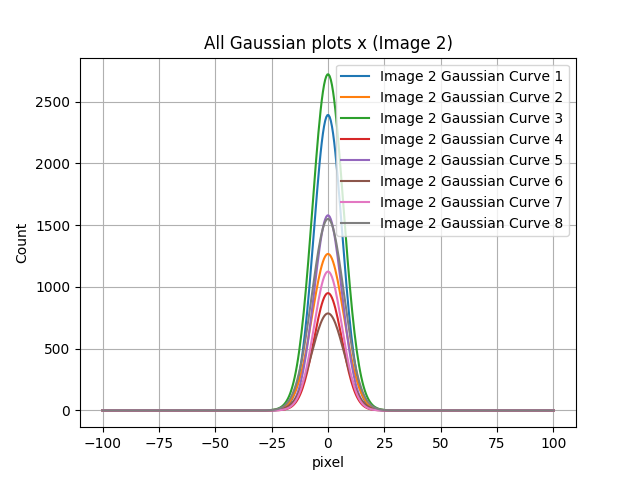
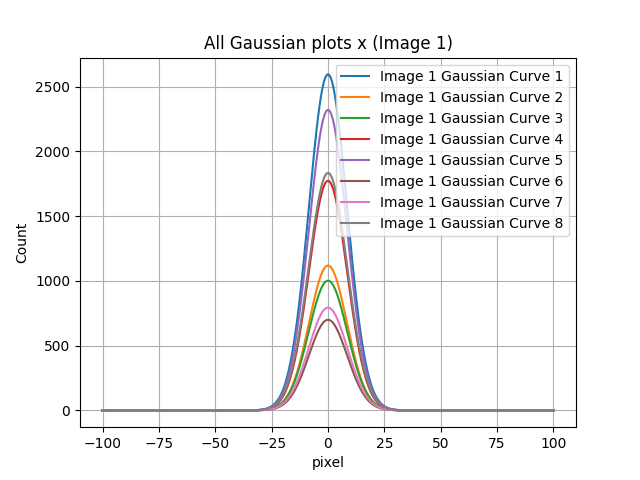
شکل ۴. داده های برازش گاوسی

که واحد موقعیت ها و سیگما ها پیکسل و واحد زاویه ها رادیان است.



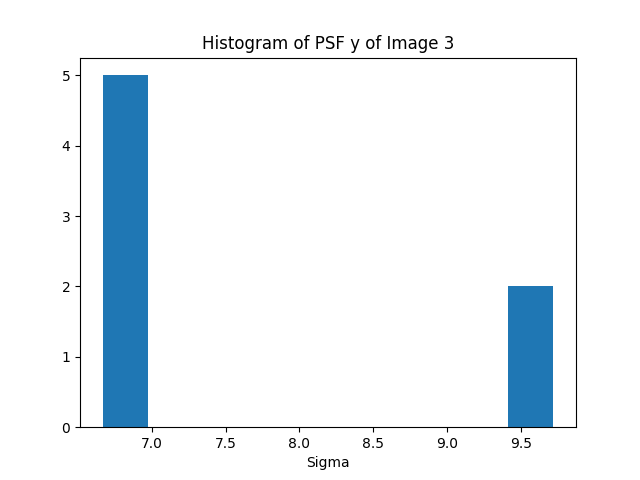
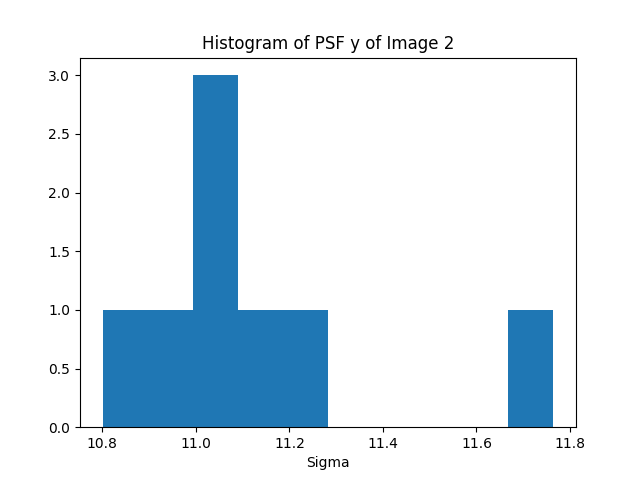
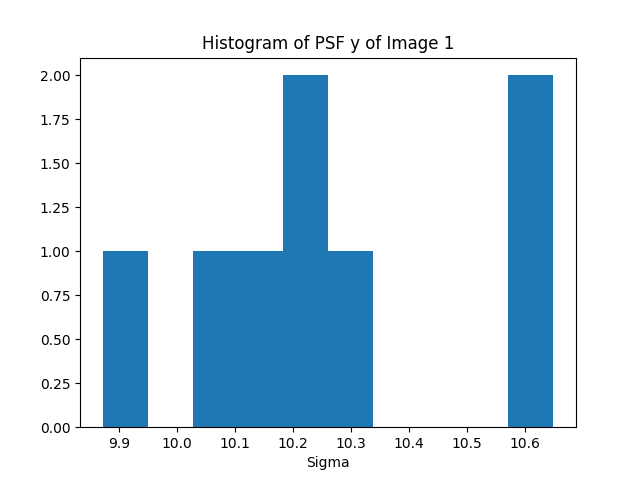
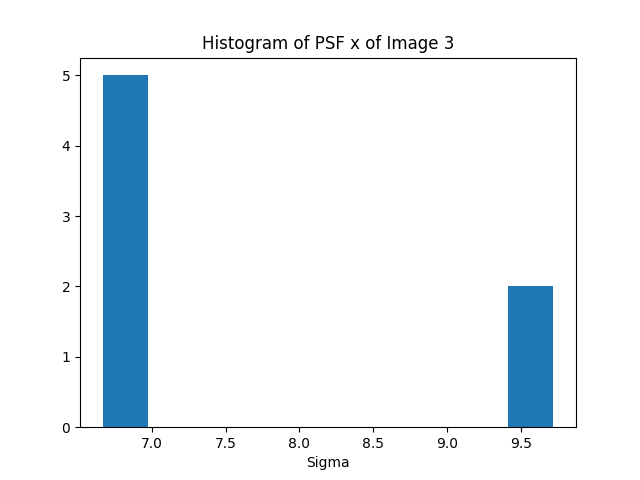
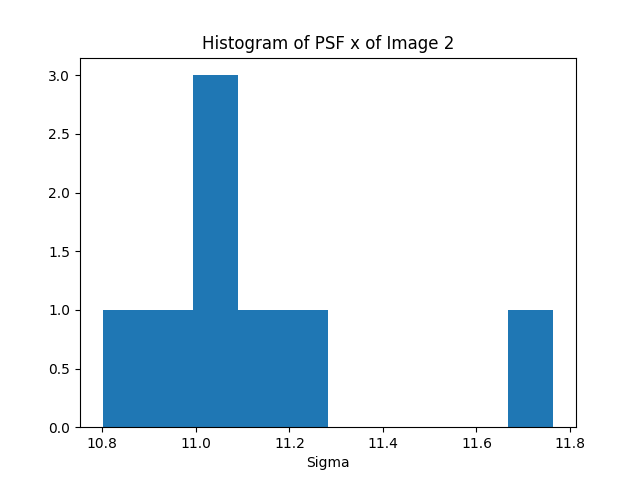
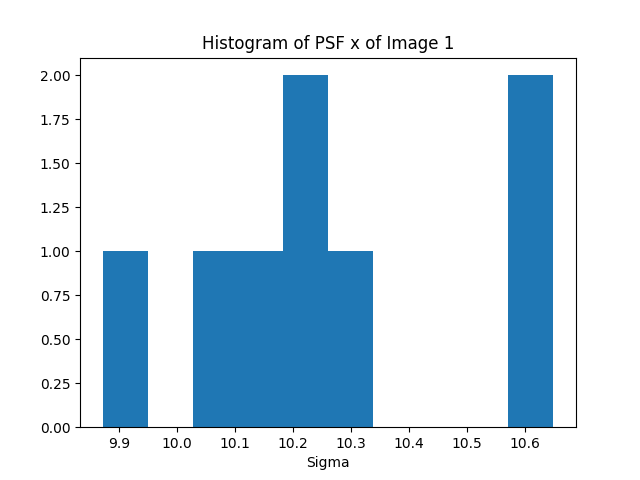
شکل ۵. نمودار تابع گاوسی دو ستاره. به دلیل زیاد بودن ستارگان تنها دو نمودار را گذاشتم.

همچنین نمودار گاوسی ستارگان هر عکس به صورت زیر است:



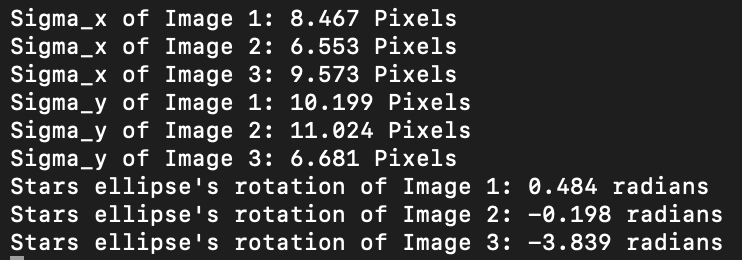
اشکال ۶ تا۱۱. نمودار های گاوسی های ستارگان عکس ها ۱ تا ۳

هیستوگرام های سیگما ها به صورت زیر است:



شکل های ۱۲ تا ۱۷. هیستوگرام های سیگما ها

همینطور داده های میانه های سیگما و چرخش منحنی گاوسی به صورت زیر است:



شکل ۱۸. داده های نهایی PSF