



UNIVERSIDAD  
**COMPLUTENSE**  
MADRID

TRABAJO FIN DE GRADO

---

**Reconocimiento automático de  
galaxias y estrellas en imágenes  
astronómicas.**

---

*Autor:*

Víctor RODRIGO  
GUDIEL

*Director:*

Carlos GREGORIO  
RODRIGUEZ

25 de septiembre de 2016

## **Resumen**

Con la irrupción de la tecnología digital de captura de imágenes, la astronomía incrementó notablemente tanto el número de imágenes que se obtiene del espacio como la calidad de las mismas.

Este conocimiento en formato digital requiere de sistemas automatizados de tratamiento e interpretación de las mismas para obtener información científica relevante, tal como extracción de fuentes puntuales, análisis de espectroscopía, calibrado, ...

El proyecto tiene como objetivo la detección y clasificación de elementos luminosos tales como estrellas y galaxias, tomando como fuente ficheros astronómicos FITS o imágenes astronómicas genéricas.

Proponemos desarrollar un software con interfaz gráfica, que permite un manejo sencillo de las funcionalidades implementadas.

Palabras clave:

Tratamiento digital de imagen. Astronomía. Detección de elementos astronómicos. Visión artificial. OpenCV. Estrellas. Galaxias. Detección de contornos. FITS. Software de procesado de imágenes.

## **Abstract**

With the advent of digital imaging technology , astronomy has significantly increased the number of images obtained from space and the quality of them.

This knowledge in digital format requires automated processing systems and interpretation thereof to obtain relevant scientific information, such as extraction of point sources, spectroscopy analysis, calibration, ...

The project focuses on the detection and classification of lighting elements such as stars or galaxies, using as source FITS files or generic astronomical images.

The project is accompanied by a software with graphical interface that allows easy management of the implemented features.

Keywords:

Digital image processing. Astronomy. Astronomical element detection. Artificial vision. OpenCV. Starts. Galaxies. Contour Detection, FITS. Image processing software.

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Imágenes Astronómicas científicas . . . . .	4
1.2. Imágenes astronómicas amateur . . . . .	7
<b>2. Objetivos</b>	<b>9</b>
2.1. Objetivos Generales . . . . .	9
2.2. Objetivos Específicos . . . . .	9
<b>3. Plan de trabajo</b>	<b>13</b>
3.1. Análisis . . . . .	13
3.1.1. Antecedentes . . . . .	13
3.1.2. Dificultades encontradas . . . . .	13
<b>4. Desarrollo</b>	<b>15</b>
4.1. Decisiones de diseño: la interfáz de usuario . . . . .	15
4.2. Prueba de los algoritmos sobre una muestra de imágenes . . . . .	15
4.3. Arquitectura interna de la aplicación . . . . .	15
4.4. Bibliotecas externas . . . . .	16
4.5. Bibliotecas propias . . . . .	18
4.5.1. Ajuste de rango dinámico . . . . .	18
4.5.2. Búsqueda de puntos de interés, estrellas y galaxias . . . . .	22
4.5.3. Eliminación de ruido y Point Spread Function (PSF) . . . . .	27
4.6. Procesado de imágenes . . . . .	28
4.6.1. Pre-procesado base . . . . .	28
4.6.2. Algoritmo de clasificación . . . . .	31
4.7. Interfaz gráfica . . . . .	34
<b>5. Conclusiones</b>	<b>39</b>
5.1. Cumplimiento de objetivos . . . . .	39
5.2. Trabajo futuro . . . . .	40
<b>6. Agradecimientos</b>	<b>42</b>

## 1. Introducción

Gran parte del conocimiento del universo deriva de la medida de fones que plasmamos en forma de imágenes, que más tarde son analizados y estudiados.

Este proyecto nace de la necesidad de automatizar la detección de cuerpos celestes en imágenes raster, concretamente, los cuerpos celestes que se busca detectar y categorizar son aquellos que emiten luz, estrellas y galaxias. Los algoritmos han de ser precisos y el software ha de tener un enfoque generalista accesible a los astrónomos no profesionales y estar sujeto a licencia de código abierto.

No solo se observa el universo desde la tierra, la humanidad ha posicionado observatorios fuera de la estratosfera, estos factores junto con el avance y abaratamiento de la tecnología ha incrementado el número de adquisiciones de imágenes astronómicas, estas imágenes en muchos casos no llegan a ser analizadas debido al tiempo humano necesario para procesarlas. La irrupción de la imagen digital y su avance en las últimas tres décadas abre el camino al tratamiento automático de las imágenes.

El tratamiento automatizado de imágenes astronómicas requiere de tecnologías tales como la visión artificial, campo que avanza rápidamente en ámbitos como la detección facial o seguimiento de objetos, pero que por las peculiaridades de las imágenes astronómicas, no ha progresado del mismo modo, los motivos entre otros son la baja definición de los elementos a tratar, los diferentes rangos dinámicos existentes, las características propias de los múltiples elementos de captación de imágenes, la falta de adecuación de los actuales algoritmos de visión artificial, ...

Los observatorios y algunas instituciones académicas se encargan parcialmente de la labor del tratamiento de las imágenes que obtienen, realizando un pre-procesado automático denominado reducción, de los datos obtenidos desde una imagen RAW o cruda a unos datos que permita su uso en un contexto científico muy específico. Esta reducción cumple los requisitos de la misión o del grupo de estudio que preparó la captura de las imágenes. El cometido de la reducción es doble:

- Modificar los datos obtenidos teniendo en cuenta las características propias del instrumento usado, aberraciones en los espejos (primario, secundario, ...), corrección de posibles defectos no previstos en la instrumentación o desgaste a lo largo del tiempo.

- Añadir otros datos (metadatos) tales como tiempo de exposición, posición en el espacio, filtros usados, etc.

Dando como resultado un fichero válido para su posterior análisis.

Los grandes proyectos astronómicos destinan la mayor parte de su presupuesto al desarrollo de instrumentación dejando sólo una pequeña parte para el desarrollo software, lo que deja abierta una puerta a nuevos avances en el tratamiento de imágenes.

Los pequeños observatorios obtienen igualmente imágenes, normalmente usando equipos de sofisticación media-baja que constan de un CCD adosado a un telescopio terrestre, estas imágenes carecen de metadatos astronómicos más allá de los que proporciona el detector.

En cualquiera de los casos podemos usar sus imágenes para obtener información no intrínseca mediante técnicas de visión artificial que determinen la existencia de elementos de interés tales como galaxias o estrellas y que además proporcionen una categorización de los mismos.

El elemento base de trabajo serán imágenes de contenido astronómico que hemos considerado interesante dividir en dos grupos según su procedencia.

## 1.1. Imágenes Astronómicas científicas

El formato por excelencia dentro de la astronomía no amateur es el conocido como formato FITS, acrónimo de Flexible Image Transport System y estandarizado en el año 1981, que ha evolucionado a la par que los instrumentos astronómicos con diferentes revisiones del standard. Una de las ventajas que nos ofrece este formato es la retro-compatibilidad con revisiones anteriores.

Es un standard abierto usado para el almacenamiento, la transmisión y procesado de datos astronómicos. A diferencia de otros formatos, FITS se diseñó para contener datos de carácter científico lo que le permite incluir información descriptiva sobre fotometría y calibración espacial.

Cada fichero consta al menos de una cabecera en formato ASCII, esta cabecera contiene pares de valores que nos permiten conocer de forma estructurada y directa (al requerir únicamente un visor de fichero de texto ASCII) información tal como el origen de la imagen, fecha de obtención, lista de filtros usados, instrumento usado, configuración del instrumento, tamaño de pixel, etc.

Este formato puede contener información mas allá de una imagen raster,

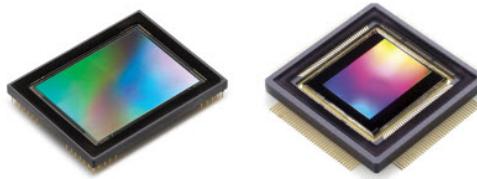


Figura 1: Sensor CCD 210W y Sensor CMOS 210W.

tales como espectros (uni-dimensionales), dataCubes (multi-dimensionales), o incluso datos estructurados (muti-tablas). Estos datos se almacenan dentro del fichero FITS en formato binario.

Las características que diferencia este formato de los formatos de imagen genérico son:

- Cada fichero contiene al menos un sistema de coordenadas propio, relacionado con la posición y rotación en el espacio desde la que fueron adquiridos los datos. El sistema de coordenadas no tiene porque ser lineal, sino que conserva las características propias del instrumento usado.
- El tamaño de la matriz que contiene la imagen es de un tamaño arbitrario, respetando la forma del detector.
- El valor que almacena cada elemento de la imagen no está limitado a 8, 16 o 32 bits, ya que lo que se almacena suele ser un conteo de fotones, el tipo habitual es de naturaleza de punto flotante.

De especial interés es el elemento encargado de la transformación de fotones al medio digital, habitualmente para esta trasformación se usa un detector CCD (charge coupled device) o raramente un detector CMOS (complementary metal oxide semiconductor).

En general los telescopios (tanto los terrestres como los situados fuera de la atmósfera) constan de una o varias ruedas de filtros (en la figura 2 situada en la página 6, podemos observar una rueda de filtro de rango infrarrojo perteneciente al telescopio espacial Hubble).

1. El color en las imágenes astronómicas:

La información de imagen que almacenan los ficheros FITS carece de información de color, ya que el valor contenido en cada celda que contiene la imagen, se corresponde con la cuenta de fotones que impactan



Figura 2: Sensor Rueda Filtros infrarrojos perteneciente al telescopio Hubble.

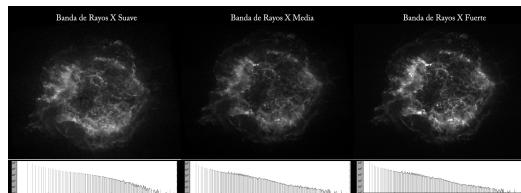


Figura 3: Imagenes procedentes de Chandra usando diferentes filtros sobre un mismo apuntado

en un determinado tiempo de exposición a una determinada longitud de onda.

Las imágenes en color que generan las instituciones son el producto de varias (en general 3 o más imágenes en escala de grises). El proceso consiste en asignar un tono a cada imagen y mezclar el resultado en una única imagen. Cada imagen ha sido capturada anteponiendo al detector un filtro físico que deja pasar determinadas longitudes de onda.

## 2. Detectores profesionales:



Figura 4: Composición de imagen en color usando trés imágenes de intensidades, cada imagen se asocia a un canal del espectro RGB

El motivo por el cual las imágenes de instrumentos astronómicos profesionales carecen de color, no es otro que el color - o rango visible - aporta muy poca información al ámbito de la astronomía. Es mucho más preciso y aporta más información, obtener un conteo del impacto de fotones en una determinada longitud de onda que una imagen en el espectro visible.

Otra de las ventajas es que se simplifica la creación de elementos de detección de imagen. Un detector que captura una imagen en rango visible RGB necesita de tres sensores por cada celdilla, lo que incrementa su coste y reduce su fidelidad al tener que colocar tres dispositivos de captura por cada celda. Esto hace que inevitablemente desviaciones en la localización exacta de los apuntados o puntos luminosos.

En contra nos encontramos con imágenes que carecen del alto impacto visual de las que gozan las imágenes en color.

## 1.2. Imágenes astronómicas amateur

Fuera del ámbito académico o técnico, lo habitual es obtener imágenes astronómicas en formatos de imagen genérico tales como TIFF, PNG , o JPG, siendo este último el menos adecuado (el algoritmo de compresión Jpeg destruirá valiosa información de luminancia tal y como se aprecia en la figura 5).

Estos formatos carecen de información astronómica en sus metadatos y lo más importante, su profundidad de color está generalmente limitada entre 8 bits (paletas indexadas con 256 posibles valores) y 32 bits (8 bits por canal), dividiéndose entre los tres colores base (RGB); con lo que se obtiene (en los casos de 24 y 32 bits)  $2^8$  intensidades diferentes, muy lejos de los rangos que manejaremos en el formato FITS.

Estas imágenes son capturadas usando tecnología de consumo general, en la mayoría de los casos sensores CCD RGB.

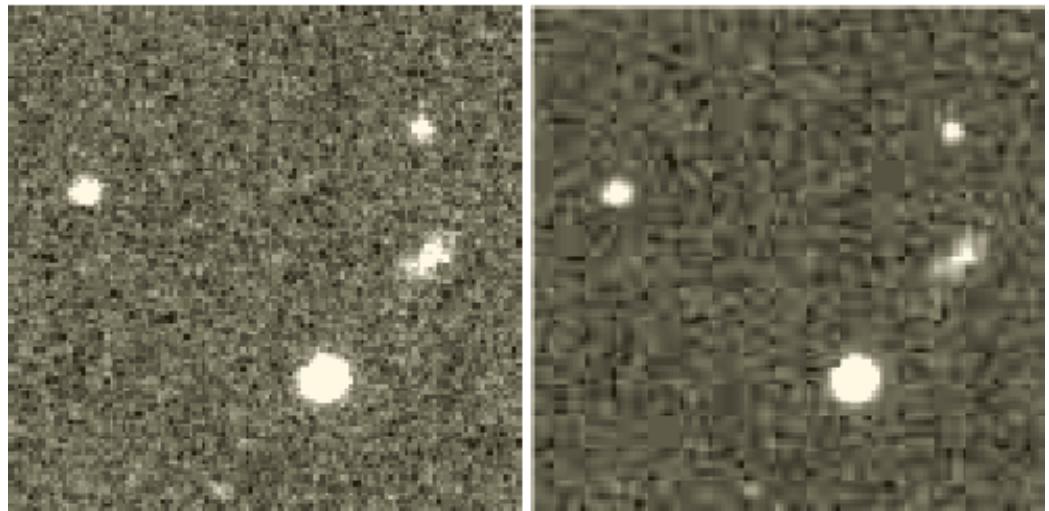


Figura 5: Efecto de la compresión jpg, a la izquierda la imagen original, a la derecha la imagen comprimida.



Figura 6: Telescopio amateur de detección de tránsitos Schmidt-Cassegrain de 8 pulgadas, consta de un CCD de 765x510 pixels

## 2. Objetivos

En este trabajo se busca explorar nuevos métodos de análisis de imágenes astronómicas, creación de nuevos algoritmos de obtención de elementos de interés o adaptación de los ya existentes.

Además de los algoritmos se ha desarrollado una interfaz de fácil manejo en la que se muestre tanto el resultado final como parte del proceso intermedio que se realiza, pudiendo variar los parámetros de configuración de los algoritmos y por lo tanto el resultado final.

El desarrollo se ha planteado de tal forma que está abierto a nuevas funcionalidades, se ha prestado especial atención en que el trabajo realizado, sirva de base futuros algoritmo de aprendizaje automático o machine-learning.

### 2.1. Objetivos Generales

El objetivo principal es la creación de algoritmos que, dada una imagen que representa una sección del espacio exterior, determinen la existencia de estrellas y galaxias.

Tras determinar la lista de puntos de interés en la imagen, el software catalogará los objetos mediante la exploración de las vecindades de los puntos destacados y haciendo uso de los algoritmos de visión por computador y criterios estadísticos.

### 2.2. Objetivos Específicos

Llevar a cabo todo el proceso implica una serie de pasos los cuales se describen a continuación:

- **Lectura de imágenes genéricas:**

Implica la lectura de formatos genéricos de imágenes raster y en caso de poseer información de color (multidimensional), trasformar esta información a un espacio de color unidimensional.

$$G : \mathfrak{R}^2 \rightarrow \mathfrak{K}^2$$

Siendo G la transformación de un espacio de dimensión 2 del cuerpo  $\mathfrak{R}$  a un espacio de dimensión 2 del cuerpo de los enteros  $\mathfrak{K}$  acotado  $[0, 2^8]$ .

- **Lectura y calibración automática de imágenes en formato FITS:**

El software ha de ser capaz de transformar el contenido de imagen del fichero FITS a un array de numPy

Tras la lectura se procede al proceso de calibración de la imagen así como una transformación del espacio de color:

$$F : \Re^2 \rightarrow \Re'^2$$

La función  $F$  se encargará de descartar los valores menores que cero (posibles errores en el detector o guardado del fichero) en  $\Re^2$ .

En nuestro caso  $\Re^2$  contiene valores no acotados, cuyo máximo y granularidad dependen del instrumento con el que fue capturada la imagen.  $\Re'^2$  almacenará la calibración de la imagen en base a la equalización elegida, bien automáticamente o bien por el usuario.

- **Detección de zonas que representen gases o polvo:**

Las estrellas por su naturaleza, iluminan sus proximidades, permitiendo ver el polvo o gas que las rodea; cuando un cúmulo de estrellas tienen en sus proximidades polvo o gases en suspensión, éstos se ven iluminados. A estas zonas difusas de las imágenes las denominaremos **halos**.

El halo galáctico es un componente extendido, más o menos esférico que se extiende en las proximidades de una galaxia (para más información ver [1]).

La distinción entre el halo y el cuerpo de la galaxia queda claramente definido en las galaxias espirales (ver figura 7), donde la forma esférica del halo contrasta con el disco plano o **flat-disc**. En las galaxias elípticas esta distinción no es clara(ver figura 8), ya que no hay una transición marcada entre el cuerpo de la galaxia y el halo.



Figura 7: Galaxia espiral NGC5457 con halo. Imagen cortesía de la E.S.A.

Una característica que usarán nuestros algoritmos, es analizar el gradiente local de los halos alrededor de los puntos de máxima intensidad (ver figura 9) para determinar la existencia de galaxias.

- **Eliminación de la función de dispersión de punto (PSF):**

La función de dispersión de punto es un fenómeno que provoca que, dado un punto lumínico y un receptor electrónico, se den dispersiones de los fotones a celdas adyacentes del detector (ver figura 10).

Un buen detector tendrá una menor PSF que un mal detector.

Eliminando la función de dispersión del punto obtendremos unos resultados que se aproximen más a la realidad.

- **Detección del espacio vacío:**

Para un correcto tratamiento de la imagen y facilitar la detección de elementos de interés mediante técnicas de suavizado y análisis por sectores, se determina un rango de color y nivel de ruido que posteriormente se eliminará de la imagen.

- **Detección de estrellas puntuales y galaxias:**

Las galaxias son aglomeraciones de estrellas, cuerpos celestes y materia cósmica localizadas en una zona del espacio por efecto de la fuerza de la gravedad y que tienen un movimiento por el espacio en conjunto.

En las inmediaciones de cada estrella perteneciente a una galaxia, es habitual encontrar polvo o gas atrapado por las fuerzas de gravitación existentes (ver figuras 8 y 9).

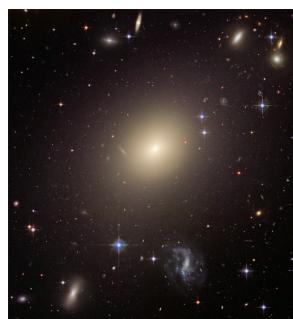


Figura 8: Galaxia elíptica ESO 325-G004 (en el centro) con halo. Imagen cortesía de NASA/ESA

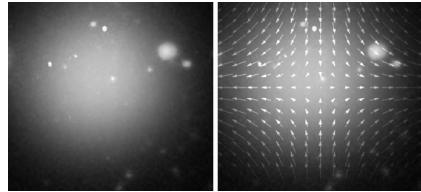


Figura 9: Cluster de Galaxias CL J1001+0220, a la derecha detalle del gradiente.  
Imagen cortesía de NASA/CXC/CEA/T

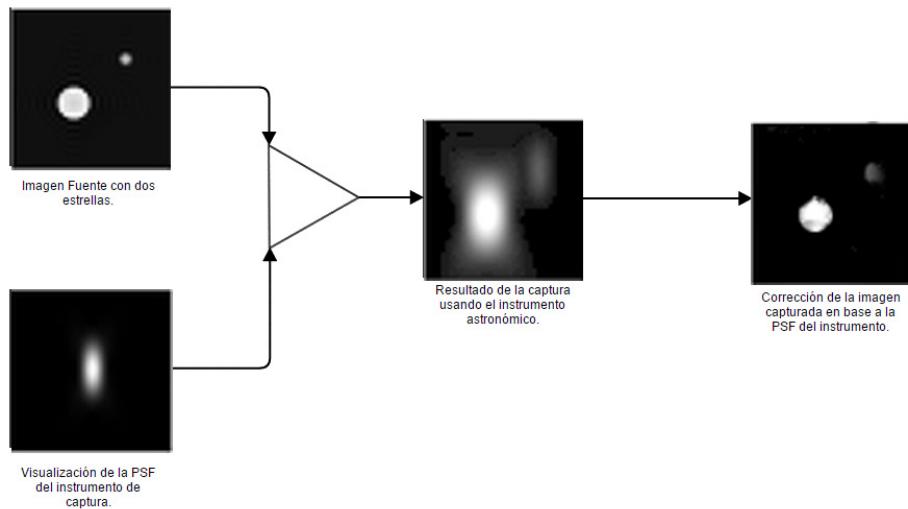


Figura 10: Corrección de PSF o función de dispersión de punto

Las estrellas que no pertenecen a una galaxia se denominan Stellar Outcast[10] ó Intergalactic Stars que al no pertenecer a una galaxia no son capaces de atraer individualmente polvo o nubes de gas.

Aprovechando el gradiente generado por las nubes de polvo y gas, el algoritmo se encarga de distinguir entre estrellas y galaxias.



Figura 11: Stellar Outcast SDSS J090745.0+24507

## 3. Plan de trabajo

### 3.1. Análisis

En esta sección se muestra casos conocidos de éxito de aproximaciones similares al ejercicio tratado y dificultades detectadas para la resolución del problema que va a ser tratado.

#### 3.1.1. Antecedentes

El único proyecto que hemos encontrado de similares características es el proyecto Galaxy Zoo.

Galaxy Zoo(<https://www.galaxyzoo.org/>) comenzó su andadura en 2007. Es un proyecto online de astronomía que invita a sus miembros a ayudar a clasificar alrededor de un millón de galaxias. Es un ejemplo de ciencia ciudadana ya que utiliza la ayuda de gente anónima para ayudar a la investigación científica.

Gran parte del software astronómico hace uso de meta-datos para localizar y dar una clasificación de estrellas y galaxias en función a una base de datos. Nuestro objetivo es *evitar la dependencia de la interacción humana o el uso de bases de datos*, supliéndola con técnicas de visión artificial.

#### 3.1.2. Dificultades encontradas

Las peculiaridades de las imágenes astronómicas hacen que los algoritmos existentes en las librerías clásicas de visión por computador no se adapten a lo esperado, esto hace que las imágenes tengan que ser transformadas. Esta transformación no es trivial y cada caso puede presentar peculiaridades.

Destacamos entre otras, las siguientes dificultades:

##### 1. Calibración de la imagen:

La calibración de las imágenes FITS a un rango dinámico correcto sobre el cual las librerías de visión artificial puedan operar.

##### 2. Baja resolución de los puntos de interés:

Los objetos de interés no tienen por qué estar centrados en la imagen, en muchos casos ocupan regiones inferiores a 16x16 pixels.

**3. Ausencia de algoritmos avanzados en la detección de objetos celestes sobre imágenes raster:**

Los algoritmos de catalogación hacen uso de las coordenadas astronómicas presentes en la cabecera del fichero FITS para cruzar estas coordenadas con bases de datos en las que están determinados los astros de interés; nuestro objetivo no es posicionar objetos de interés en base a los datos almacenados en una base de datos sino descubrir posibles objetos de interés en una imagen.

**4. Dificultad para conseguir ficheros de prueba:**

Las bases de datos astronómicas existentes contienen grandes cantidades de imágenes cuya forma de acceso se realiza mediante sus coordenadas astronómicas; dando como resultado de la búsqueda una larga lista de imágenes FITS.

Además el peso de las mismas hace difícil su gestión.

## **4. Desarrollo**

### **4.1. Decisiones de diseño: la interfáz de usuario**

Queremos crear una interfaz de trabajo, la interfáz gráfica ha de tener un doble propósito:

1. Facilitar la configuración de los algoritmos y visualizar el histograma resultante para la correcta automatización.  
En un futuro trabajo se pretende hacer uso de técnicas de aprendizaje automático. La interfaz serviría como punto de inicio para obtener los parámetros base necesarios para el aprendizaje automático.
2. Permitir al usuario no especializado trabajar con sus imágenes y obtener el resultado de forma visual.

Sería deseable que la interfaz visual sea agradable en apariencia y de fácil manejo, mostrando las opciones de manipulación que necesita el usuario y los datos más significativos para nuestros algoritmos.

Se valoraron diferentes tecnologías (QT en sus variante PyQt y PySide), optando por TkInter por su versatilidad.

Para aprovechar lo mejor posible el area de ventana se habilitará una zona con pestañas en la que se mostrará la imagen original y procesada (para más detalle ver subsección 4.7).

### **4.2. Prueba de los algoritmos sobre una muestra de imágenes**

Para determinar la validez de los algoritmos, sería deseable disponer de imágenes que incluyen tanto imágenes FITS como imágenes genéricas.

Se partirá de una pequeña batería de imágenes, sobre la cual se aplicarán los algoritmos que se desarrollarán (para más detalle ver sección 4.5), de este modo podremos afinar su precisión. Esta fase ha permitido realizar la automatización del proceso afinando los algoritmos de clasificación.

### **4.3. Arquitectura interna de la aplicación**

Todo el desarrollo se ha centrado en el uso del lenguaje Python: es un lenguaje maduro con una gran cantidad de bibliotecas de apoyo. Tanto las

bibliotecas usadas como el interprete de Python son de código abierto.

La aplicación se divide en dos ficheros:

- `ide.py`: contiene la creación de la interfaz gráfica, el flujo que sigue el tratamiento de cada imagen y la parte encargada de la automatización. Se ha usado un enfoque orientado a objetos.
- `cvSpace.py`: en este fichero se encuentran las funciones encargadas del análisis y tratamiento de las imágenes, funciones que hemos dividido en dos grupos: el grupo de la parte estadística y el grupo del tratamiento de imágenes.  
Este fichero usa un enfoque imperativo.

Hacemos especial hincapié en el uso de código abierto, ya que dentro de los objetivos del proyecto está la creación de un conjunto de herramientas que puedan ser usadas, modificadas y copiadas libremente por la comunidad; por ello, todo el código se ha desarrollado bajo la licencia BSD.

#### 4.4. Bibliotecas externas

**OpenCV** Es una librería de visión artificial, desarrollada bajo la licencia BSD y creada inicialmente por Intel. Es multiplataforma con su núcleo en C y tiene envolturas para gran cantidad de lenguajes, entre ellos Python. Es la librería más avanzada que existe para visión artificial, destacando por su optimización que permite la realización de aplicaciones en tiempo real. OpenCV es una librería madura (se inicio en 1.999) sobre la que gran cantidad de proyectos se apoyan.

**NumPy** Numpy es el paquete base fundamental para cálculos científicos en el entorno de Python. Al usar Numpy podemos expresar las imágenes como matrices multidimensionales. Esta representación de las imágenes en arrays de NumPy no solo es eficiente desde el punto de vista computacional sino también en la gestión de recursos y nos abre la posibilidad de conectar con otras librerías que usan NumPy. Sus puntos fuertes son:

- Capacidad de manejo de matrices de datos genéricos que usaremos para guardar y manipular nuestras imágenes.
- Alta velocidad en las operaciones que se realizan sobre las matrices.

- Multitud de funciones matemáticas implementadas de serie.
- NumPy es la base de otras librerías, como por ejemplo scikit-learning, librería de aprendizaje automático.

La licencia de NumPy es BSD.

**Matplotlib** Librería de dibujo en 2D/3D que destaca por su facilidad de manejo y potencia. Es la encargada de mostrar los histogramas en la interfaz gráfica.

La licencia de Matplotlib es BSD.

**Astropy** Astropy es un proyecto comunitario para acercar a python al mundo de la astronomía. Su núcleo está desarrollado en C y Python. El proyecto Astropy busca crear un framework robusto con las funcionalidades generales que se requieren en astronomía. Sus desarrolladores y usuarios principales son profesionales de la astronomía y al ser un proyecto comunitario cubre muchos de sus ámbitos (fotometría, spectrografía, sistemas de coordenadas, etc).

La principal funcionalidad que usamos de Astropy es un módulo que nos permite la carga de ficheros FITS.

La licencia de astropy es BSD.

**TkInter** Es la librería de facto ( y más usada ) para desarrollar interfaces gráficas en python. Se basa en TcL/Tk lo que le permite ser multiplataforma y crear interfaces nativas sobre el sistema operativo en el que se esté ejecutando. Toda la interfáz gráfica se apoya en esta librería.

La licencia de Tkinter es Python License.

## 4.5. Bibliotecas propias

El desarrollo englobado en esta sección está directamente relacionado con los objetivos propuestos, partiendo de la lectura de imágenes genéricas como de imágenes en formato astronómico FITS, así como los algoritmos que se encargarán del preprocesado necesario para disponer de una imagen transformada que finalmente será procesada por los algoritmos de clasificación. El resultado de esta secuencia de procesos nos dará una clasificación de los elementos de interés.

Los algoritmos desarrollados los hemos englobado en el fichero cvSpace.py cuyo cometido es realizar todas las tareas de Visión por computador que nos planteamos en los objetivos y estadística; se destacan:

### 4.5.1. Ajuste de rango dinámico

Su principal cometido es trasladar el rango de intensidades de cada elemento de la matriz imagen a unos rangos normalizados. De especial interés en este

#### 1. Lineal

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz, realiza la siguiente transformación dando como resultado la figura 12.

$$Pixel'[x, y] = \frac{2^8 * (Pixel[x, y] - m)}{(M - m)}$$

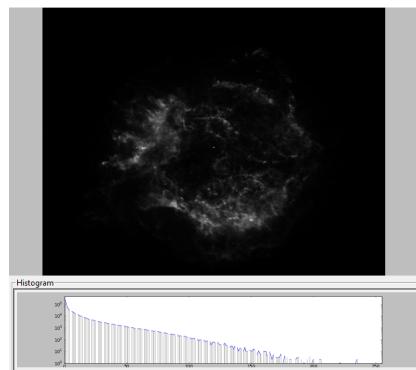


Figura 12: Ajuste de rango dinámico lineal

## 2. Raíz cuadrada

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz, realiza la siguiente transformación (Ver figura 13): El resultado del valor del nuevo pixel se almacena en  $\text{Pixel}'[x, y]$ .

$$\text{Pixel}'[x, y] = \frac{2^8 * (\text{Pixel}[x, y] - m)}{\sqrt{M - m}}$$

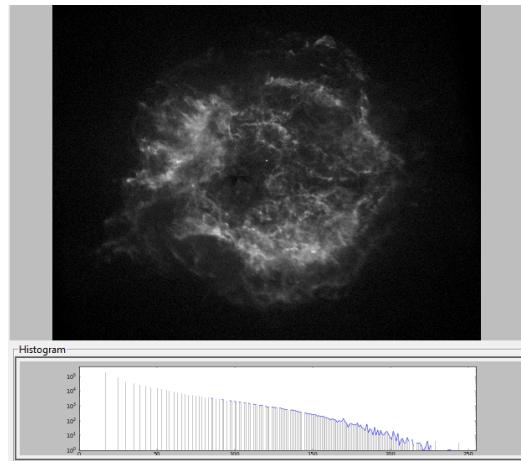


Figura 13: Ajuste de rango dinámico Raíz Cuadrada

## 3. Logaritmo

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz, realiza la siguiente transformación (Ver figura 14):

$$\begin{aligned} \text{Factor} &= \log_{10}(M - m) \\ \text{Pixel}'[x, y] &= \frac{\log_{10}(\text{Pixel}[x, y])}{\text{Factor}} \end{aligned}$$

## 4. Potencia

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz y dado un exponente  $n$ , realiza la siguiente transformación (Ver figura 15):

$$\begin{aligned} \text{Factor} &= \frac{1}{(M - n)^n} \\ \text{Pixel}'[x, y] &= (\text{Pixel}[x, y] - m)^n * \text{Factor} \end{aligned}$$

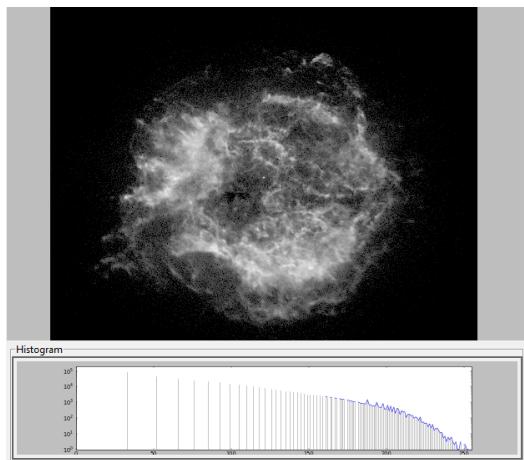


Figura 14: Ajuste de rango dinámico logarítmico

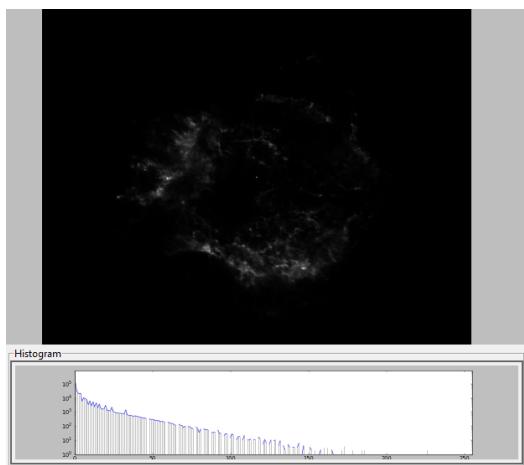


Figura 15: Ajuste de rango dinámico potencia con valor 1.5

## 5. Arcoseno Hiperbólico

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz y dado un factor de no linealidad  $nL$ , realiza la siguiente transformación (Ver figuras 16 para  $nL=1.5$  y 17 para  $nL=5$ ):

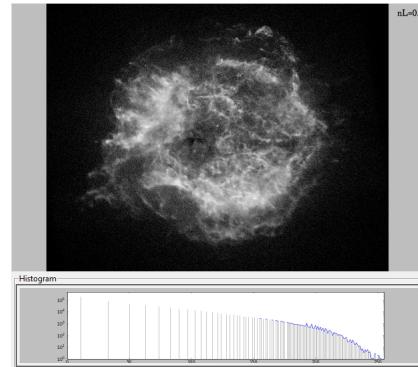


Figura 16: Ajuste de rango dinámico ArcoSeno hiperbólico con valor 1.5

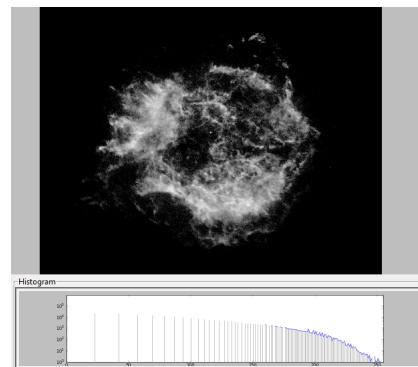


Figura 17: Ajuste de rango dinámico ArcoSeno hiperbólico con valor 1.5

$$\text{Factor} = \frac{\text{arcSinH}((M-m)}{nL}$$

$$\text{Pixel}'[x, y] = \frac{\frac{\text{arcSinH}(\text{Pixel}[x, y]-m)}{nL}}{\text{Factor}}$$

## 6. Ajuste en base al histograma

La ecualización del histograma de una imagen es una transformación que pretende obtener un histograma de la imagen con una distribución uniforme. Es decir, que exista el mismo número de pixels para cada nivel de gris del histograma de una imagen monocroma, ocupando todo el rango. (Ver figura 18).

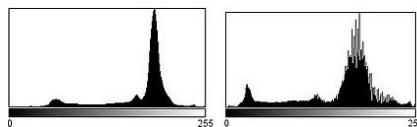


Figura 18: Ajuste de rango dinámico por ecualización de histograma; a la derecha histograma original, a la izquierda histograma equalizado.

### 4.5.2. Búsqueda de puntos de interés, estrellas y galaxias

En este punto se detallan las funciones creadas para obtener el objetivo del trabajo, la determinación de objetos de interés.

Varias funciones son las encargadas de buscar, descartar y catalogar las posibles estrellas y galaxias.

Destacamos por su interés *getObjectList*, *getGalaxyCenter* y *getcontours*.

#### 1. *getObjectList*

La función *getObjectList* presente en *cvSpace.py* hace uso de la técnica detección de blobs de openCV para localizar puntos de interés, concretamente los textitcandidatos a estrellas.

La búsqueda mediante blobs<sup>1</sup> se centra en la variación de la intensidad de cada pixel con su vecindad; el proceso en detalle realiza las siguientes operaciones:

- a) Convierte la imagen a una imagen binaria.
- b) Extrae las componentes conectadas en base a un cálculo de contornos.

---

<sup>1</sup>Un blob es una región de la imagen que difiere en sus propiedades como color o intensidad, con respecto a sus vecindades

- c) En caso de blobs muy cercanos entre sí, los agrupa en un único blob. Este proceso es de especial interés para la agrupación de estrellas que conforman una galaxia.
- d) Se eliminan aquellos blobs que son detectados de forma errónea; para ello se hace uso de la imagen de espacio vacío generada mediante el método `generateDarkImage` de la clase `AstroImage`. Se descartan aquellos candidatos cuyo centro de coordenadas ha caído sobre la zona marcada como espacio vacío, conservándose el resto (ver figura 19).

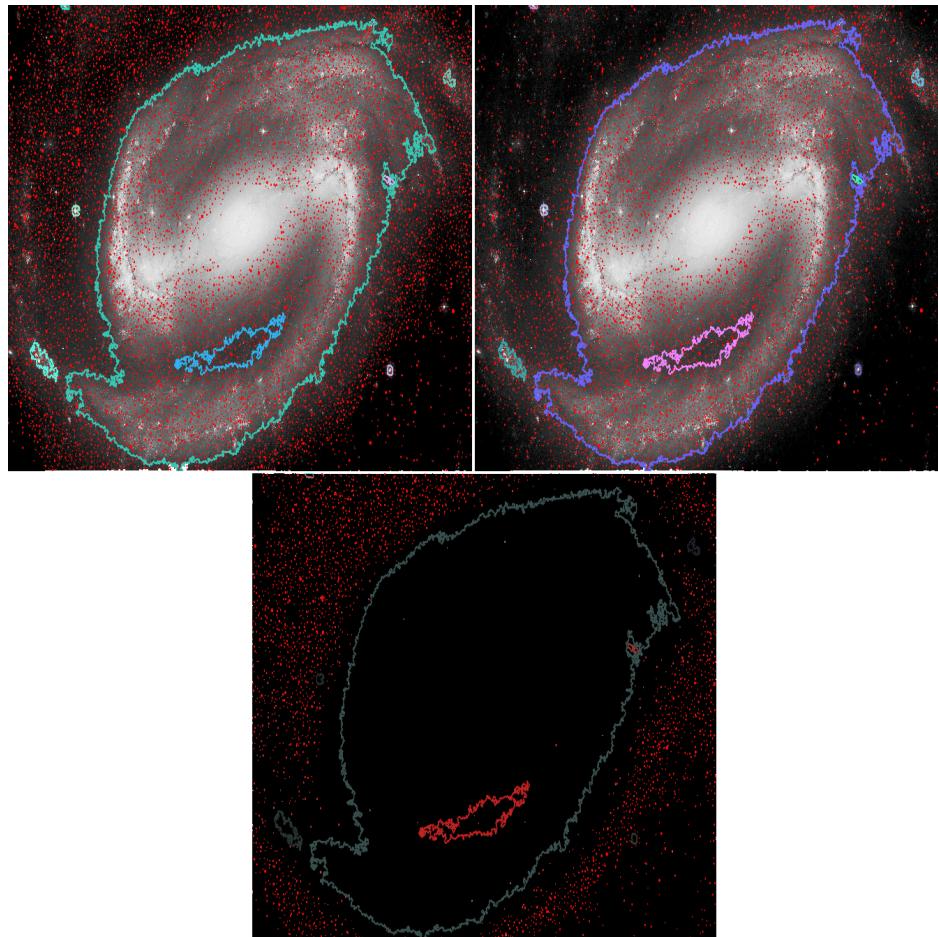


Figura 19: Resultado del proceso de categorización y eliminación de puntos de interés erróneos.

- e) Para cada grupo de blobs nos devuelve tanto su centro como su área.

## 2. `getGalaxyCenter`:

La función `getGalaxyCenter` se encarga de determinar el centro de las candidatas a galaxias.

El proceso que realiza `getGalaxyCenter`, se resume en los siguientes pasos:

- a) Aplicación de blur a la región de interés.
- b) Búsqueda del máximo local en intensidad.

Es necesario realizar un blur <sup>2</sup> o difuminado, como se aprecia en las imágenes 20 21 y 22.

En caso de no aplicar el suavizado a la imagen, el pico de mayor intensidad no tendría por qué estar situado en el centro de la galaxia. Al aplicar el suavizado, se pondera en media el valor de cada pixel con su vecindad y aumentará la probabilidad de que el centro calculado se ajuste al centro de la galaxia. Esto es debido a que en el centro de la galaxia se acumula la mayor área lumínica.

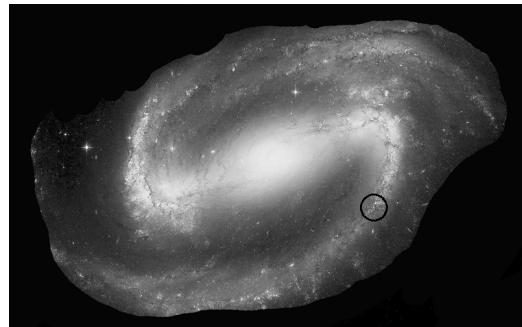


Figura 20: Obtención errónea del centro de la galaxia (marcado con una circunferencia) al aplicar la función `getGalaxyCenter` sobre la imagen sin procesar con un filtro de suavizado.

---

<sup>2</sup>[2]

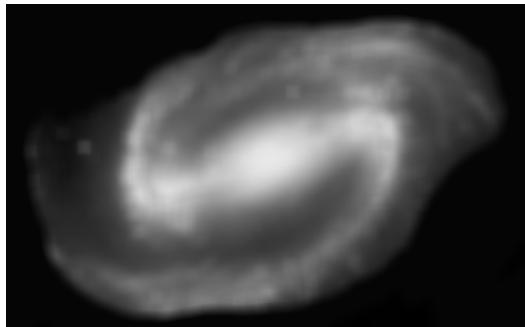


Figura 21: La imagen original ha sufrido el proceso de suavizado.

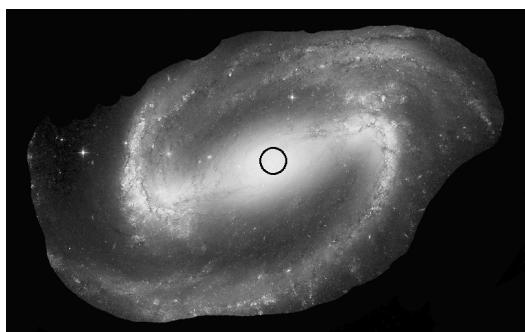


Figura 22: Obtención correcta del centro de la galaxia (marcado con una circunferencia) al aplicar la función `getGalaxyCenter` sobre la imagen procesada con un filtro de suavizado.

### 3. `getContours`:

`getContours` aplica búsquedas de contornos sobre la imagen transformada para detectar formaciones de galaxias e internamente realiza operaciones de erosión, dilatación y cierres de contornos. Bajo estas líneas se presenta el código usado.

```

def getContours(imOrig, maxContours=10, t1=200, maxCtNumber = 1000):
    maxC = maxContours
    nContour = 0
    lastContours = 0
    contours = []
    while ((2*lastContours>=nContour) and maxC>0):
        im = dilate(closeContour(imOrig, maxC-1),(maxC+1)/2)
        imgrey = im
        ret,thresh = cv2.threshold(imgrey,t1,255,0)
        lastContours = nContour
        nContour = len(contours)
        if (lastContours == 0):#caso 0
            lastContours = nContour
            maxC=maxC-1
        epsilon = 10
        contours=[cv2.approxPolyDP(ctn,epsilon,True)for ctn in contours]
        area = np.zeros(len(contours))
        isParent = np.zeros(len(contours))

        index = 0
        for cnt in contours:
            if (hierarchy[[0],[index],[3]]===-1):
                isParent[index]=1
                area[index] = cv2.contourArea(cnt)
                index=index+1
        meanArea = np.mean(area)
        isSubContour = np.where(isParent==0)
        if (len(contours)>maxCtNumber):
            toDelete = np.where(area<=.5*meanArea)
        else:
            toDelete = np.where(area<=.05* meanArea)
        goodContours = np.delete(contours, toDelete)
    return goodContours, meanArea

```

#### 4.5.3. Eliminación de ruido y Point Spread Function (PSF)

1. Eliminación de ruido: Mediante heurística y en base al resultado del algoritmo de búsqueda de contornos, se eliminan áreas carentes de interés y producto del ruido.

El algoritmo realiza cálculos incrementales del número de contornos, estimando el ruido por el crecimiento a cada iteración. Si de una iteración a la siguiente el número de contornos crece en un factor de más del doble, se estima que se ha llegado a un umbral correcto.

La eliminación del ruido en nuestro caso es una tarea compleja, ya que la diferencia entre una región de ruido y una región que contiene una estrella resulta difícil hasta para un ojo entrenado.

2. Eliminación de PSF:

Aplicando operaciones morfológicas sobre la imagen, concretamente una dilatación seguida de una erosión, se eliminan los puntos detectados erróneamente como válidos.

El parámetro clave para la eliminación de la PSF es el kernel aplicado tanto a la dilatación como a la erosión que es de un tamaño de 5. Como se aprecia en la figura (23), la forma que toma la PSF varía según el filtro físico usado al captar la imagen dese el telescopio, presentando formas cercanas a una Normal N-Dimensional. Un kernel es una matriz que se aplica a cada pixel de la imagen.

Expresando este procedimiento en una ecuación tenemos:

$$K(i, j) = \frac{1}{25} * I$$

$$H(x, y) = \sum_{i=0}^{M_i-1} \sum_{j=0}^{M_j-1} I(x + i - a_i, y + j - a_j) * K(i, j)$$

El resultado de aplicar un kernel sobre una imagen es una nueva imagen con las mismas dimensiones que la imagen de origen (ver figura ??), pero con posibles valores distintos en cada pixel; estos valores dependen tanto del kernel elegido como de las características de la imagen.

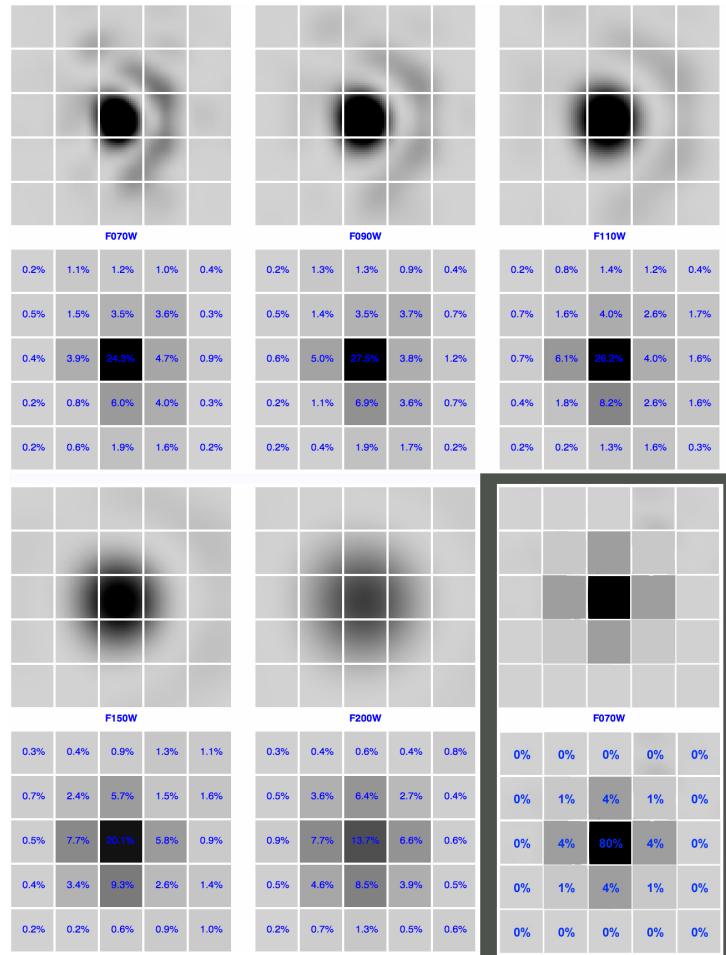


Figura 23: Distintas PSF (5) de un instrumento astronómico al aplicar diferentes filtros físicos de longitud de onda. La última imagen (inferior derecha) se corresponde con la simulación de reducción de PSF

## 4.6. Procesado de imágenes

### 4.6.1. Pre-procesado base

El siguiente esquema detalla el proceso que sigue cada imagen y al que llamaremos pre-procesado base. (Ver figura 24)

En el caso de la carga de una imagen astronómica FITS se realiza un ajuste del rango dinámico, este paso se obvia para las imágenes en formato raster

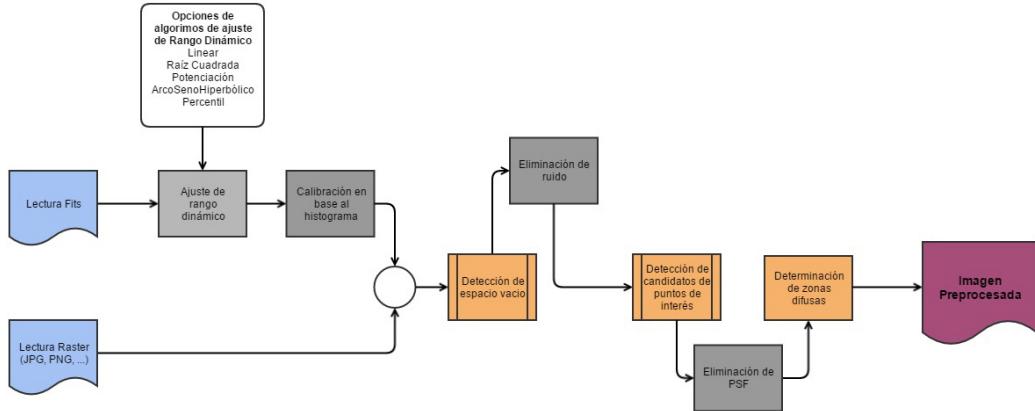


Figura 24: Flujo de pre-procesado

genérico ya que se presupone que el instrumento de captura tiene por salida una imagen en el rango esperado.

La imagen FITS se convierte a una matiz de 8 bits, cuyo contenido dependerá de la ecualización aplicada (Ver figura 25).

Se calculan los valores del histograma (ver figura 26) que definen los um-

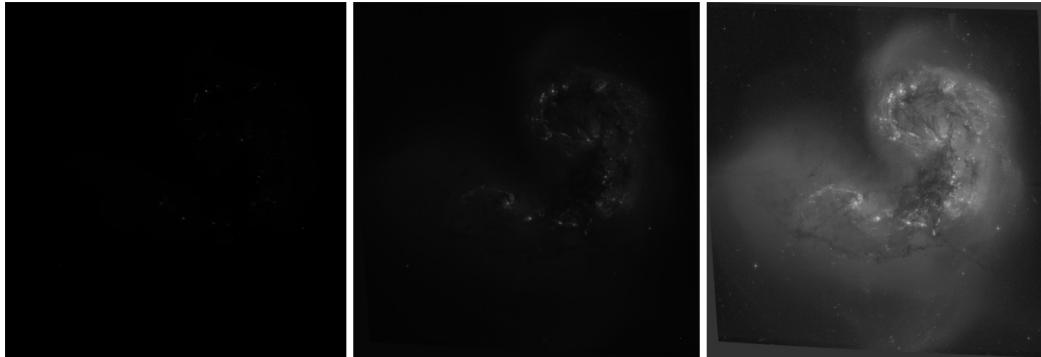


Figura 25: De izquierda a derecha, imagen ecualizada linealmente, aplicando raíz cuadrada y aplicando arco seno hiperbólico.

brales con los que se generan tres matrices o imágenes, que denominaremos imagen Espacio vacío, imagen Halo e imagen Candidatos.

Estas imágenes o matrices (ver figura 27), serán usadas en el paso de clasifi-

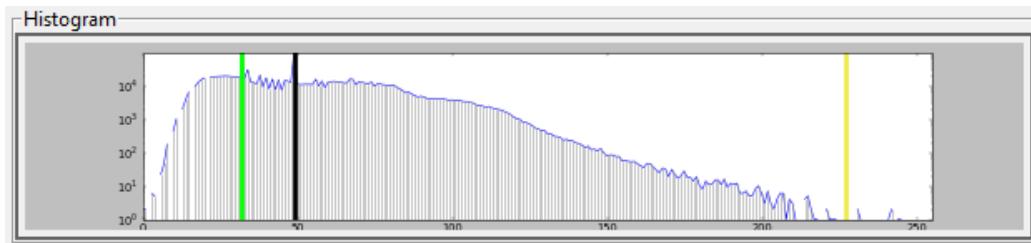


Figura 26: Lineas de ecualización sobre histograma.

cación.

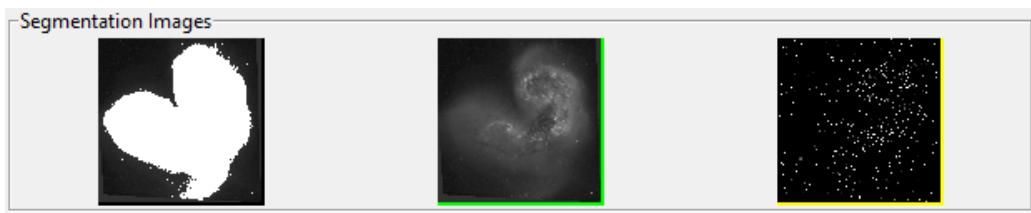


Figura 27: Imágenes o matrices autogeneradas que se usarán para la catalogación.

#### 4.6.2. Algoritmo de clasificación

Partiendo de una imagen adaptada por el pre-procesado base, se realizan los siguientes pasos que se detallan en el esquema.(Ver figura 28)

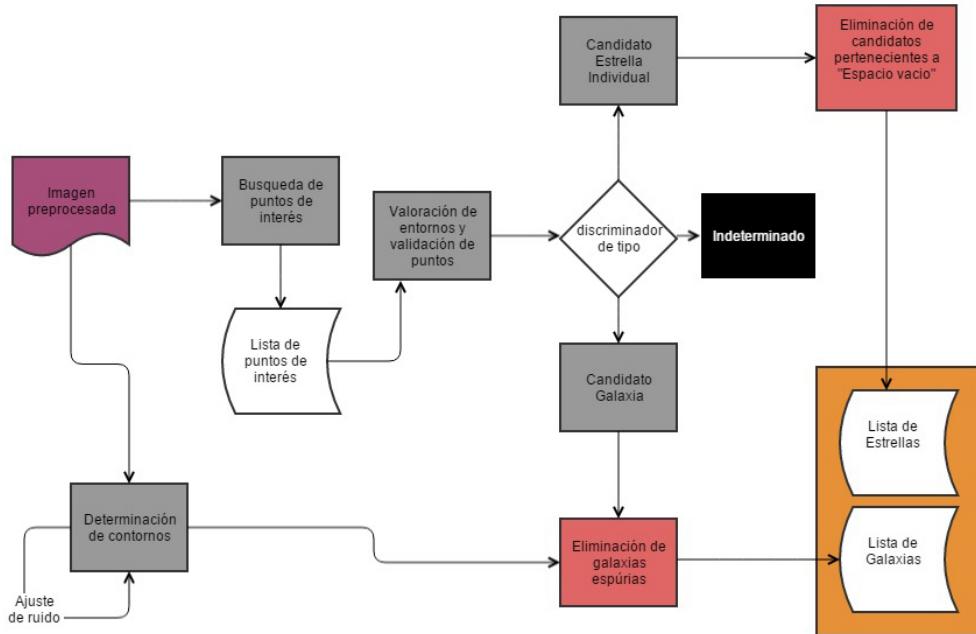


Figura 28: Flujo de clasificación

#### Obtención de coordenadas de puntos de interés:

Se genera una lista de candidatos partiendo de la imagen Candidates obtenida tras el pre-procesado.

Para cada coordenada candidata se valorará, mediante su entorno (contenido en la imagen Halo), si es un posible candidato a estrella o galaxia. Ciertos puntos se descartarán por ser información espuria (probablemente formados artificialmente en el detector). Como resultado obtendremos una imagen en la que vienen marcados los puntos catalogados como estrellas y las galaxias resaltadas por su contorno (ver figura 29)

#### Clasificación de estrellas:

Para la clasificación de estrellas se usa una aproximación múltiple, primero la detección en base a un umbral (todo conjunto de puntos que tienen intensidad superior a ese umbral pasan a ser candidatos) y segundo la búsqueda

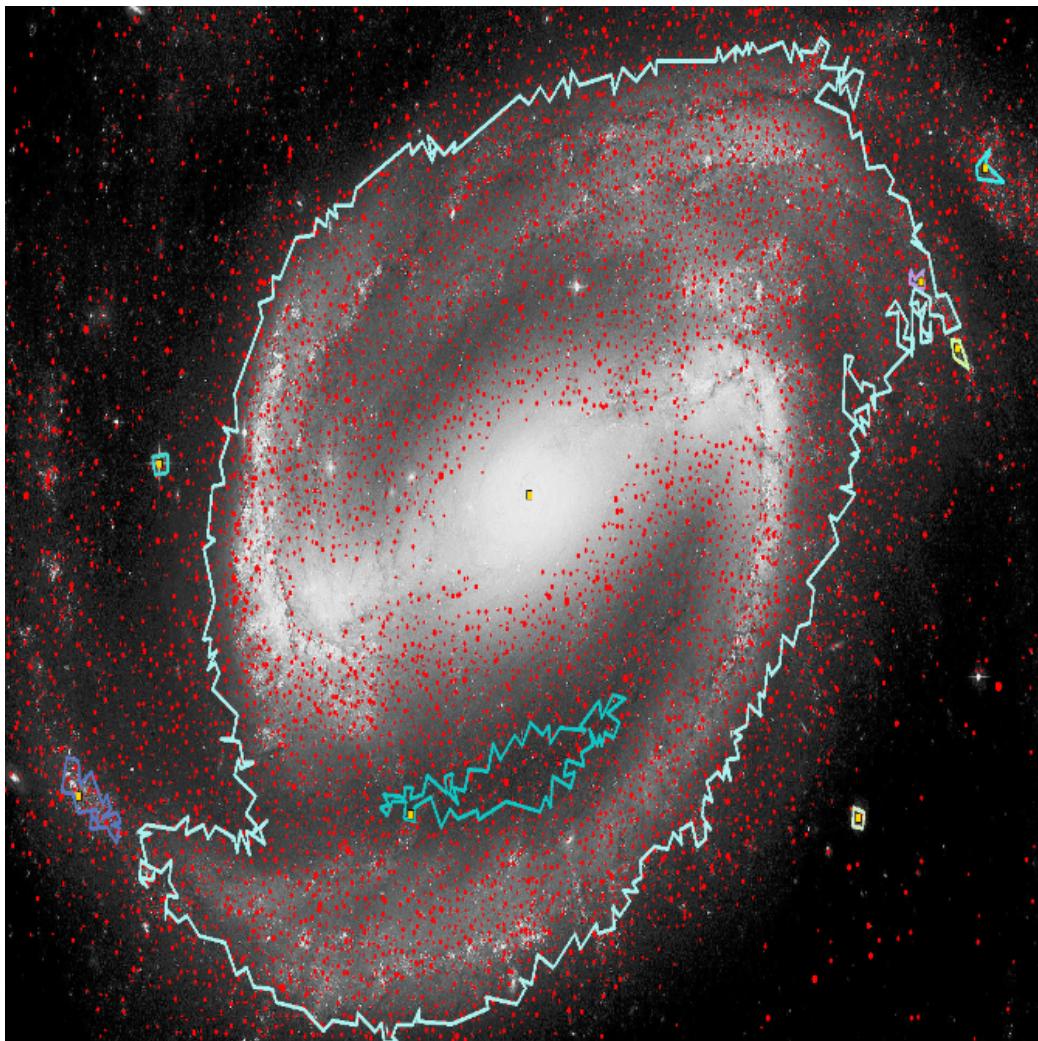


Figura 29: Resultado de la clasificación. Marcado en rojo parecen las estrellas, los contornos denotan las galaxias.

de blobs usando OpenCV.

Con los datos generados de ambas aproximaciones se hace un último filtrado que elimina el posible ruido catalogado como candidato. Para ello se hace uso de la imagen-matriz de zonas oscuras, eliminando los puntos que recaen en zonas catalogadas como espacio vacío o ruido.

#### Clasificación de galaxias:

Para la clasificación de galaxias se hace un uso intensivo de búsqueda de contornos, que una vez obtenidos, en base al área media y a su número, se realiza un proceso de eliminación de contornos que no representan galaxias. La parte encargada de la eliminación hace uso de herramientas estadísticas sencillas tales como la media y varianza.

## 4.7. Interfaz gráfica

Tanto para el usuario final como para la validación de los algoritmos en tiempo real se ha creado una interfaz (Ver figura 30) que permite: visualización de la imagen, ajuste de una serie de parámetros y presentación visual de los elementos astronómicos detectados.

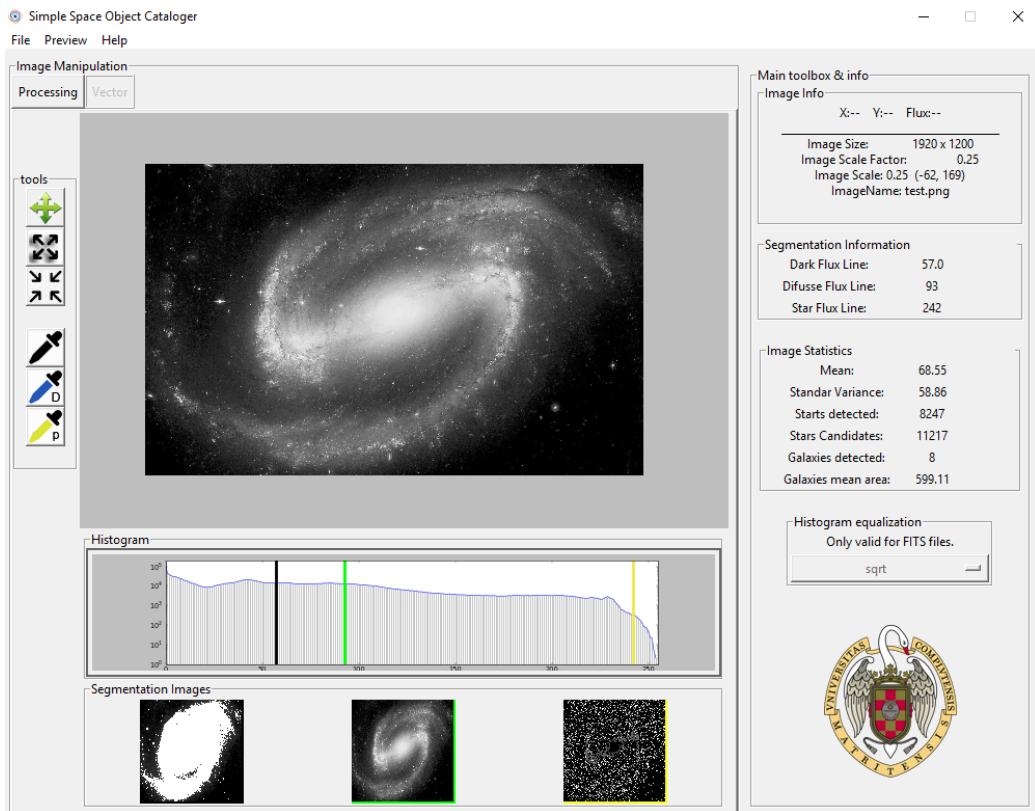


Figura 30: Interfáz gráfica

La interfaz gráfica se compone de los siguientes elementos

- Menú superior:

Tiene por secciones File, Preview y Help.

File: las opciones que se permiten son la carga de imágenes, el guardado del proyecto y la conversión y guardado (si procede) de una imagen

FITS (ver figuras 31 y 32).

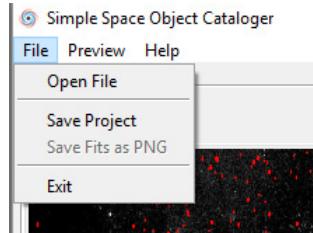


Figura 31: Menú Superior con una imagen raster cargada, la opción Save Fits as PNG aparece desactivada.

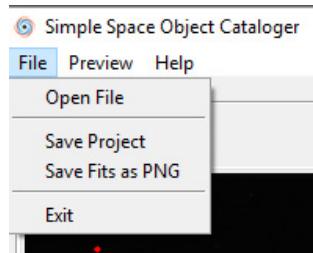


Figura 32: Si cargamos una imagen FITS se activa la opción de convertir y guardar en formato PNG.

Preview: Contiene la opción Invert Image, que invierte la imagen mostrada, esta opción es útil para apreciar mejor (en algunos casos) las diferencias de color (ver 33).

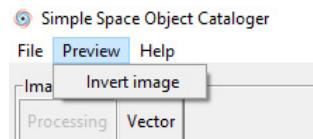


Figura 33:

Help: Sólo contiene una sección About en la que se informa de la versión del software y dónde descargarlo (ver figura 34).

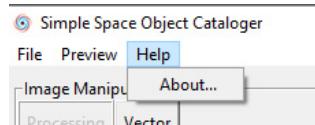


Figura 34:

- Zona central:

Muestra el área central de procesado y tiene dos sub-secciones: la sección processing y la sección vector.

La sección Processing es el área principal de trabajo, se muestra la imagen astronómica en tratamiento, el histograma y las sub-imágenes generadas (ver imagen 35) en base a la imagen cargada.

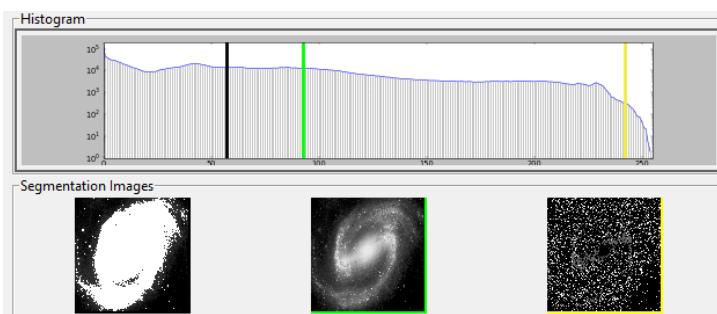


Figura 35: Histograma e imágenes segmentadas

Contiene también herramientas de navegación (mover, aumentar y reducir la imagen), así como selectores de intensidad para modificar la segmentación de las imágenes (ver imagen 36). Cada pipeta se corres-



Figura 36: Herramientas de navegación y selectores de segmentación

ponde con una línea de color del histograma, siendo la primera pipeta en color negro, la que se encarga de manejar la línea de intensidad del espacio vacío, la pipeta de color verde la que maneja la línea de intensidad del halo y la pipeta amarilla la encargada de manejar la línea de intensidad pico relacionada con las estrellas.

Al pulsar sobre cualquiera de las pipetas, el cursor cambiará, permitiendo sobre la imagen de trabajo pulsar en cualquier punto para modificar los valores del procesado.

La sección Vector es el resultado del proceso, en ella se muestra sobre la imagen cargada y procesada, las estrellas y zonas de galaxias encontradas (ver figura 37).

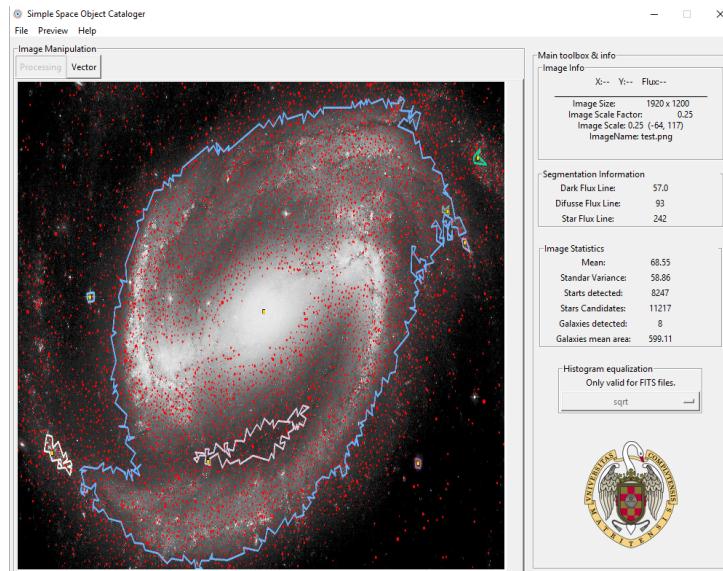


Figura 37: Sección Vector: Resultado del procesado y catalogación.

- Elementos de información y control de equalización (ver figura 38): En la parte derecha de la ventana está presente el cuadro de información que contiene datos estadísticos de la imagen tales como su media y varianza, el número de estrellas o galaxias detectadas y la información del pixel sobre el que tenemos el puntero del ratón.

En la sección inferior está el selector de ecualización, que nos permite ver cómo se transforma la imagen en función de la ecualización que

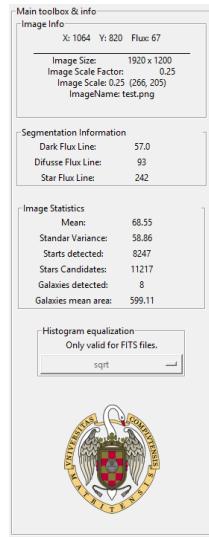


Figura 38:

apliquemos. Esta opción está únicamente disponible cuando el fichero fuente es FITS.

## 5. Conclusiones

El desarrollo de este trabajo ha resultado ser un medio de aprendizaje y adquisición de nuevos conocimientos.

También ha sido una búsqueda de soluciones a los errores y problemas que se presentaban.

Trabajar con OpenCV me ha hecho conocer una librería excelente, su wrapper de python sorprende por su sencillez y velocidad.

EL uso de la biblioteca TkInter ha resultado ser muy flexible, permitiéndome un desarrollo rápido y usable con poco esfuerzo.

Se ha conseguido, partiendo de una idea muy ambiciosa, llegar a un prototipo que integra la mayoría de funcionalidades.

### 5.1. Cumplimiento de objetivos

El resultado del proyecto es un software funcional, rápido y manejable que cumple con los objetivos básicos planteados y en el que se deja la puerta abierta a mejoras y nuevas funcionalidades que quedaron descartadas debido a la complejidad que presentaban en función del tiempo con el que contaba. Cada una de las metas propuestas inicialmente cumple su cometido.

Los problemas que se han presentado durante la realización han sido:

- Durante el desarrollo se han tenido que descartar propiedades tales como la clasificación detallada de la morfología de las galaxias, la complejidad es superior a la estimada inicialmente, y para obtener un prototipo con las máximas funcionalidades, se han pospuesto.
- Las peculiaridades de la imagen astronómica hacen difícil estimar ciertas características. Un ejemplo claro es el siguiente:  
Las imágenes fuente pueden incluir, en el caso de pertenecer a alguna misión astronómica, zonas sobre-expuestas, esto es debido a que el aparato de captación fue configurado para obtener datos de un punto poco luminoso, haciendo que el resultado sea una imagen que contiene

sobre-exposición en zonas más luminosa. Este es uno de los muchos inconvenientes con los que el software se encuentra a la hora de calibrar la imagen y obtener las líneas de intensidad sobre, datos con los que se realiza todo el proceso.

## 5.2. Trabajo futuro

Tal y como se ha comentado anteriormente, se ha dejado una puerta abierta a mejoras:

- Los ajustes heurísticos requieren afinamiento, no se descarta tomar nuevas estrategias, por ejemplo en la obtención de los contornos de los candidatos a galaxias establecer una relación entre el área y el perímetro para de este modo detectar estructuras que presenten crestas o pliegues (ver en figura 39, página 40)

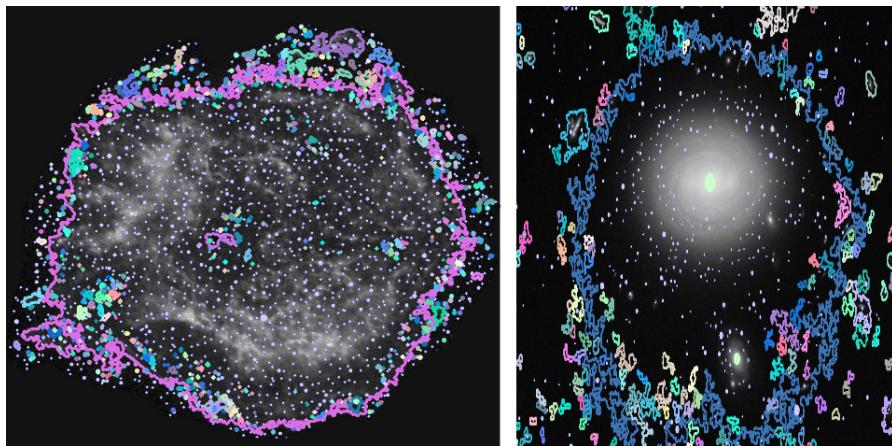


Figura 39: El comportamiento de los contornos presenta excesivos pliegues en la imagen de la derecha (tono azul) en comparación con el comportamiento de la imagen de la izquierda (tono violeta). El contorno de la derecha es un claro ejemplo de contorno erróneo.

- Mejoras de optimización. Un claro ejemplo es la detección de estrellas y galaxias que se pueden unificar en una única función, o la gestión de las imágenes internas que en algunos (thumbnails) casos se duplican innecesariamente ocupando temporalmente memoria.

- Añadir la funcionalidad de clasificación acorde a la secuencia de Hubble (ver figura 40), para ello, todo el proceso previo ha de ser óptimo en sus resultados.

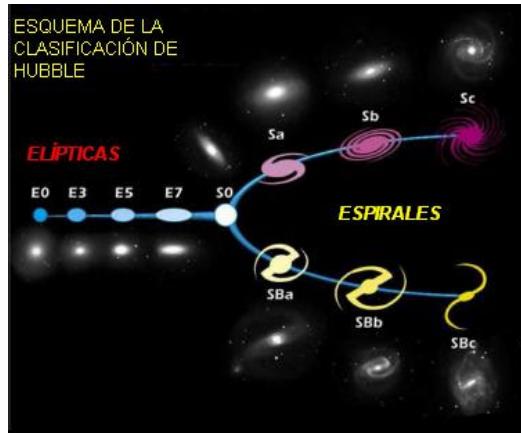


Figura 40: Secuencia del doctor Hubble para la clasificación de galaxias.

- Otro punto interesante podría ser la creación de un método que detecte los destellos de lente o lens flare (ver figura 41), ya que nuestro algoritmo los puede interpretar como galaxias, por su halo en determinados casos.



Figura 41: Lens flare en una captura producida por el telescopio Hubble de Proxima Centauri. Imagen cortesía de ESA/Hubble NASA.

## **6. Agradecimientos**

Quiero agradecer a Carlos Gregorio Rodríguez la oportunidad de realizar este trabajo fin de grado, la ayuda y guía que me ha brindado en todo momento y su apoyo en la búsqueda de soluciones y la adquisición de nuevos conocimientos.

A Miguel Angel Valero Espada por darme un enfoque distinto de lo que es el campo de la programación. Ana Inés Gomez de Castro por sus enseñanzas en Astronomía.

Muy especialmente a Rocío por animarme en todo momento y estar siempre a mi lado, así como a mi familia por su apoyo.

Sin todos ellos me hubiese resultado muy difícil el desarrollo del proyecto.

Éste trabajo está dedicado especialmente a mis yejos Mercedes y Modesto por todos los valores que desde bien pequeño me supieron transmitir.

## Referencias

- [1] BARNES, JOSHUA E.  
*Encounters of disk/halo galaxies Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 331, Aug. 15, 1988, p. 699-717.*  
08/1988  
<http://articles.adsabs.harvard.edu//full/1988ApJ...331..699B/0000699.000.html>
- [2] WILLOW GARAGE ET ALL.  
*OpenCV: Open Source Computer Vision. Documentación oficial.*  
<http://docs.opencv.org/>  
Intel's research center in Nizhny Novgorod  
2.000 (v:1.0) - 2015 (v:3.0)
- [3] GONZALO PAJARES, JESÚS M. DE LA CRUZ  
*Visión por computador: Imágenes digitales y aplicaciones*  
ISBN: 84-7897-472-5.  
RA-MA Editorial.  
2.001
- [4] JEAN-LUC STARCK, FIONN MURTAGH  
*Astronomical image and Data Analysis. (Second edition)*  
ISBN: 3-540-42885-2  
Springer-Verlag Berlin Heidelberg.  
2.006
- [5] ROBERT LUPTON.  
*SDSS Image Processing I: The Deblender.*  
<http://www.astro.princeton.edu/~rhl/photomisc/deblender.pdf>  
Princeton University Observatory  
2.001
- [6] ALI MOSLEH, J.M. PIERRE LANGLOIS, PAUL GREEN2  
*Image Deconvolution Ringing Artifact Detection and Removal via PSF Frequency Analysis.*  
<http://www.groupes.polymtl.ca/amosleh/papers/ECCV14.pdf>  
Polytechnique Montréal  
2.014

- [7] NASA/GSFC  
*The FITS Support Office - Documentación oficial formato FITS.*  
<http://fits.gsfc.nasa.gov/>
- [8] JEAN-PIERRE JESSE, VICTOR-VALERIU PATRICIU ET AL.  
*Detection and Identification of Elliptical Structure Arrangements in Images : Theory and Algorithms.*  
<http://ethesis.inp-toulouse.fr/archive/00001847/01/Patraucean.pdf>  
 Institut National Polytechnique de Toulouse  
 2.012
- [9] MICHAEL R. BLANTON ET ALL.  
*Improved background subtraction for the Sloan Digital Sky Survey images.*  
<http://arxiv.org/pdf/1105.1960.pdf>  
 2.011
- [10] DAVID A. AGUILAR, CHRISTINE PULLIAM PUBLIC  
*First Stellar Outcast Discovered by Astronomers*  
<https://www.cfa.harvard.edu/news/2005-05>  
 Harvard-Smithsonian Center for Astrophysics  
 2.005
- [11] HARVARD-SMITHSONIAN CENTER FOR ASTROPHYSICS (CFA) *Brain Tumor Detection Using Histogram Thresholding to Get the Threshold point.*  
<http://www.iosrjournals.org/iosr-jeee/Papers/Vol9-issue5/Version-3/D09531419.pdf>  
<https://www.cfa.harvard.edu/news/2005-05>  
 2.005