



# UNIVERSIDAD COMPLUTENSE MADRID

TRABAJO FIN DE GRADO

---

## Visión artificial: Reconocimiento automático de galaxias y estrellas en imágenes astronómicas.

---

*Autor:*

Víctor RODRIGO  
GUDIEL

*Director:*

Carlos GREGORIO  
RODRIGUEZ

11 de septiembre de 2016

## **Resumen**

Con la irrupción de la tecnología digital de captura de imágenes, el mundo de la astronomía incrementó notablemente el número de imágenes que se obtiene del espacio, no solo aumentó la calidad sino que se vio beneficiada de la facilidad de copia y difusión de las mismas. Este vasto conocimiento en formato digital requiere de sistemas automatizados de tratamiento e interpretación de las mismas.

El proyecto se centra en la detección y clasificación de elementos luminosos tales como estrellas y galaxias, tomando como fuente ficheros astronómicos FITS o imágenes raster genéricas y mediante técnicas de visión artificial, obtener una lista de fuentes luminosas, su posición relativa a la imagen y características descriptivas de las mismas.

El proyecto se acompaña de una simple interfaz gráfica que permita a los posibles usuarios un manejo sencillo y en cierta medida un uso didáctico.

## **Abstract**

With the advent of digital imaging technology , the world of astronomy exponentially increased the number of images obtained from space.

This vast knowledge in digital format requires automated systems processing and interpretation thereof.

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

The ultimate goal is to obtain a list of light sources , its position relative to the image and descriptive characteristics of the same, and a graphical interface that allows easy handling to potential users and maybe some didactics workflow.

# Índice

<b>1. Introduction</b>	<b>4</b>
1.1. Imágenes Astronómicas científicas . . . . .	5
1.1.1. El color en las imágenes astronómicas . . . . .	6
1.1.2. Ventajas de un detector profesional . . . . .	8
1.2. Imágenes astronómicas amateur . . . . .	8
<b>2. Objetivos</b>	<b>9</b>
2.1. Objetivos Generales . . . . .	9
2.2. Objetivos Específicos . . . . .	10
<b>3. Plan de trabajo</b>	<b>13</b>
3.1. Antecedentes: . . . . .	13
3.2. Documentación y búsqueda de algoritmos . . . . .	14
3.3. Determinación de los puntos de difícil solución . . . . .	14
3.4. Creación de una interfaz de trabajo . . . . .	14
3.5. Testeo de los algoritmos sobre un conjunto pequeño de imágenes	14
3.6. Test de fiabilidad . . . . .	14
<b>4. Desarrollo</b>	<b>15</b>
4.1. Arquitectura interna de la aplicación . . . . .	15
4.2. Bibliotecas externas . . . . .	15
4.2.1. OpenCV . . . . .	15
4.2.2. NumPy . . . . .	15
4.2.3. Matplotlib . . . . .	16
4.2.4. Astropy . . . . .	16
4.2.5. Tkinter . . . . .	16
4.3. Bibliotecas propias . . . . .	16
4.3.1. Ajuste de rango dinámico . . . . .	17
4.3.2. Búsqueda de puntos de interés . . . . .	20
4.3.3. Eliminación de ruido . . . . .	21
4.3.4. Eliminación de PSF . . . . .	21
4.3.5. Determinación de forma en las galaxias . . . . .	21
4.4. Línea de procesamiento de imágenes . . . . .	21
4.4.1. Pre-procesado base . . . . .	21
4.4.2. Algoritmo de clasificación . . . . .	22
4.5. Interfaz gráfica . . . . .	22

<b>5. Conclusiones</b>	<b>25</b>
5.1. Cumplimiento de objetivos . . . . .	25
5.2. Trabajo futuro . . . . .	25
<b>6. Agradecimientos</b>	<b>25</b>
<b>7. Referencias bibliográficas</b>	<b>25</b>

# 1. Introduction

Gran parte del conocimiento del universo deriva de la medida de fotones 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 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 abierto el campo a mejoras o nuevas iniciativas en un campo tan amplio y técnico como es la unión de la visión artificial y la astronomía.

Pequeños observatorios o particulares 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 y su definición en muchos casos es baja, buscando en muchos la belleza visual por encima de obtención de datos de uso científico, aunque conteniendo igualmente material digno de estudio.

En cualquiera de ambos 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, proporcionándonos además una categorización de estos elementos.

El elemento base de trabajo serán imágenes de contenido astronómico, 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. Uno de los requisitos y gran cualidad de 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 pensando en 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, tales como espectros (uni-dimensionales), dataCubes (multi-dimensionales),

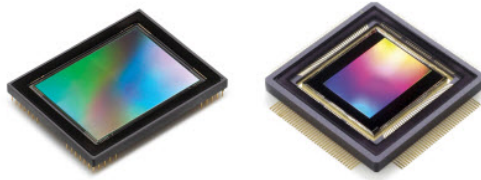


Figura 1: Sensor CCD 210W y Sensor CMOS 210W.

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 transformación se usa un detector CCD (charge coupled device) o raramente un detector CMOS (complementary metal oxide semiconductor).

En general los telescopios (tanto terrestres como fuera de la atmósfera) constan de una o varias ruedas de filtros, en la figura 2 situada en la página 7, podemos observar una rueda de filtro de rango infrarrojo perteneciente al telescopio espacial Hubble.

#### 1.1.1. 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



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

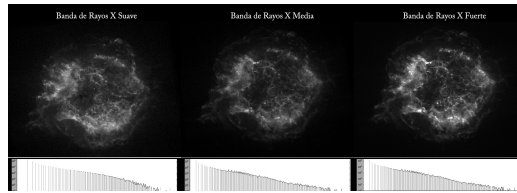


Figura 3: Imágenes procedentes de Chandra usando diferentes filtros sobre un mismo apuntado

imagen, se corresponde con la cuenta de fotones que impactan 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.

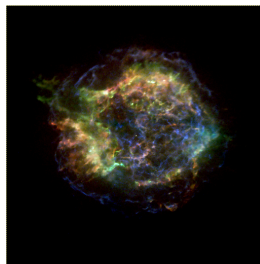


Figura 4: Composición de imagen en color usando tres imágenes de intensidad, cada imagen se asocia a un color del espectro RGB



### **1.1.2. Ventajas de un detector profesional**

El motivo por el cual las imágenes de instrumentos astronómicos no amateur 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 3 dispositivos de captura por cada celda, lo que produce 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 el menos adecuado este último (el algoritmo de compresión Jpeg destruirá valiosa información de luminancia). 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: Telescopio amateur de detección de tránsitos Schmidt-Cassegrain de 8 pulgadas, consta de un CCD de 765x510 pixel

## 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.

A parte 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 fijando las bases a un futuro algoritmo de aprendizaje automático o machine-learning.

### 2.1. Objetivos Generales

El objetivo principal es la creación de algoritmos que, dado 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 detallan escuetamente a continuación:

- Lectura de imágenes genéricas:  
Implica la lectura de formatos genéricos de imágenes raster y en caso de poseer información de color, transformar a un espacio de color unidimensional.  
$$G : \mathcal{I}^2 \times \mathcal{R} \times \mathcal{G} \times \mathcal{B} \rightarrow \mathcal{I}^2 \times \mathcal{S}$$
  
Siendo  $\mathcal{S}$  la transformación de un espacio de dimensión 3 a un espacio de dimensión 1 discreto  $[0, 2^8)$
- Lectura y calibración automática de imágenes en formato FITS:  
El software ha de ser capaz de determinar que el fichero FITS contiene una imagen, en caso de no contenerla lanzar un error al usuario.  
Tras la lectura se procede al proceso de calibración de la imagen así como una transformación del espacio de color:  
$$F : \mathcal{I}^2 \times \mathcal{R} \rightarrow \mathcal{I}^2 \times \mathcal{R}'$$
  
Siendo  $\mathcal{R}'$  la transformación de un espacio de dimensión 1 a un espacio de dimensión 1 usando diferentes técnicas según las propiedades del histograma. En este caso  $\mathcal{R}$  contiene valores no acotados, cuyo máximo y granuralidad dependen del instrumento con el que fue capturada la imagen.
- 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 se eliminará de la imagen.
- 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, una característica que usa el software para determinar la existencia de galaxias es la existencia de este halo. En su detección interviene la variación de gradiente desde puntos de alta intensidad.
- Eliminación de la función de dispersión de punto (PSF)  
La función de dispersión de punto es un fenómeno que provoca que dado

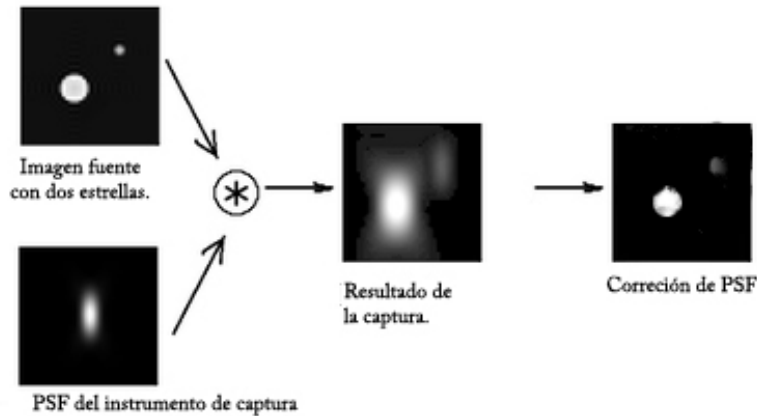


Figura 6: Corrección de PSF o función de dispersión de punto

un punto lumínico y un receptor electrónico, se den dispersiones de los fotones a celdas adyacentes del detector. Un buen detector tendrá una menor PSF que un mal detector.

Eliminando la función de dispersión del punto obtendremos unos resultados más próxima a la realidad.

- Detección de estrellas puntuales y galaxias.  
Las galaxias son aglomeraciones de estrellas que tienen un movimiento por el espacio 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 7 y 8, en la página 12)  
Las estrellas que no pertenecen a una galaxia se denominan "Stellar Outcast"ó "Intergalactic Stars", 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.
- Catalogación de galaxias en base a su morfología:  
El resultado final se centra en la obtención de un listado de galaxias junto con su clasificación en base a la clasificación dada por el astrónomo Hubble (ver figura 9, en la página 13).  
La catalogación se basa en la morfología de la galaxias, concretamente

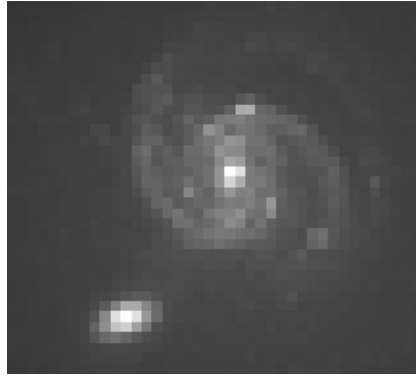


Figura 7: Galaxia con halo producido por la iluminación de polvo o nubes de gas cercanas

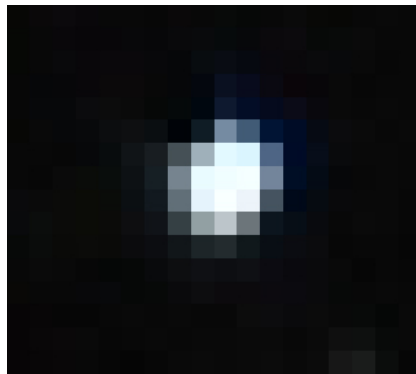


Figura 8: Estrella no perteneciente a una galaxia

divide entre elípticas y espirales y subdivide cada uno de estos grupos en base nuevamente a su morfología, para las elípticas se tiene en cuenta su excentricidad y para las espirales, su extensión relativa al punto de máximo flujo.

Dependiendo de los segundos de arco que ocupe una galaxia, el algoritmo afinará en mayor o menor medida en la clasificación.

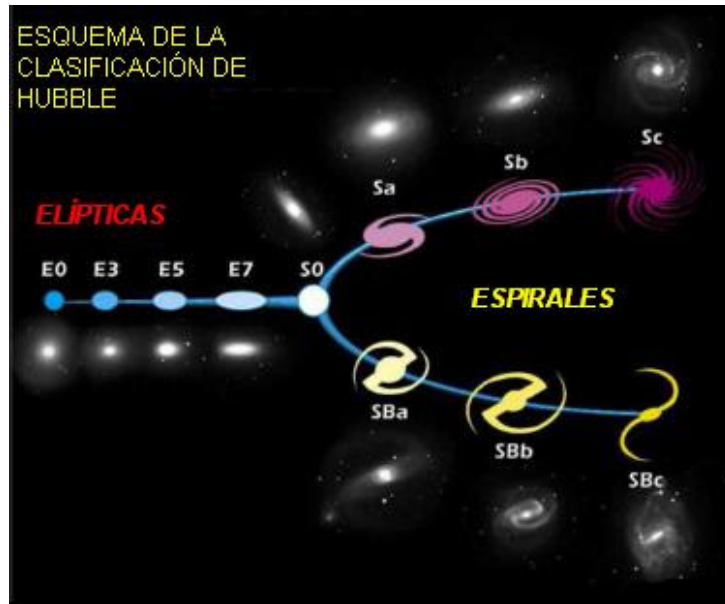


Figura 9: Esquema de clasificación del Doctor Hubble

### 3. Plan de trabajo

La línea temporal seguida es la siguiente:

#### 3.1. 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 y utiliza una enorme base de datos de imágenes, la idea detrás de este proyecto es alimentar la clasificación con la respuesta de voluntarios para llegar a determinar de forma colectiva la naturaleza de cada elemento que existe en su base de datos, usando técnicas de aprendizaje automático.

El resto de software astronómico hacen uso de los metadatos para localizar y dar una clasificación de estrellas y galaxias en función a una base de datos, nuestro objetivo es en cierta medida evitar esta dependencia y suplirla con técnicas de visión artificial.

### **3.2. Documentación y búsqueda de algoritmos**

[Redactar correctamente] Se ha buscado información tanto en el ámbito astronómico como en el de visión artificial, completar con bookmarks.

### **3.3. Determinación de los puntos de difícil solución**

Destacamos dificultades en la calibración de la imagen, la baja resolución que pueden presentar los puntos de interés, la falta de información que muchos algoritmos utilizan, como por ejemplo contar con una secuencia temporal de imágenes que permitiese discriminar el fondo y el ruido y sobre todo la calibración de las imágenes FITS a un rango dinámico correcto sobre el cual openCV pueda operar.

### **3.4. Creación de una interfaz de trabajo**

Tiene un doble propósito:

1. Facilitar la configuración de los algoritmos y visualizar el histograma resultante para la correcta automatización futura.
2. Permitir al usuario no especializado trabajar con sus imágenes y obtener el resultado de forma visual.

### **3.5. Testeo de los algoritmos sobre un conjunto pequeño de imágenes**

Para determinar la validez de los algoritmos, se ha contado con una pequeña base de imágenes, estas imágenes han sido tanto ficheros fits como imágenes genéricas.

Esta fase ha permitido realizar la automatización del proceso afinando los algoritmos de clasificación.

### **3.6. Test de fiabilidad**

Se ha ejecutado el software sobre un conjunto mayor de imágenes para comprobar su correcto funcionamiento, se adjunta tabla de resultados sobre la fiabilidad obtenida.

## 4. Desarrollo

### 4.1. 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, lo que es un punto más a favor del uso de este lenguaje.

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 desarrollado recae en la licencia BSD.

### 4.2. Bibliotecas externas

#### 4.2.1. OpenCV

Es una librería de visión artificial creada inicialmente por Intel, es multi-plataforma, su núcleo está desarrollado en C, pero 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) y una innumerable cantidad de proyectos hacen uso de la misma.

La licencia de openCV es BSD.

#### 4.2.2. NumPy

Numpy es el paquete base fundamental para cálculos científicos en el entorno de Python, al usar Numpy podemos expresar imágenes como matrices multidimensionales. Representar imágenes como arrays de NumPy no solo es eficiente desde el punto de vista computacional, sino también en la gestión de recursos, sino que 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.

#### 4.2.3. Matplotlib

Librería de dibujo en 2D/3D que destaca por su facilidad de manejo y potencia, es la encargada de mostrar en la interfaz los histogramas.

La licencia de Matplotlib es BSD.

#### 4.2.4. Astropy

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

La principal funcionalidad que usamos de astropy es el módulo fits que nos permite la carga de ficheros fits.

La licencia de astropy es BSD.

#### 4.2.5. 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.

### 4.3. Bibliotecas propias

Los algoritmos usados se encuentran en el fichero cvSpace.py, su cometido es realizar todas las tareas de Visión por computador y estadística, se dividen en:

#### 4.3.1. Ajuste de rango dinámico

Su principal cometido es trasladar el rango de intensidades de cada elemento de la matriz imagen a unos rangos normalizados, dependiendo de los datos de entrada, se elige uno u otro de los siguientes algoritmos.

Se ha tomado como referencia para mostrar los cambios una imagen FITS procedente del telescopio Chandra.

##### 1. Lineal

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz, realiza la siguiente transformación (Ver figura 10)

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

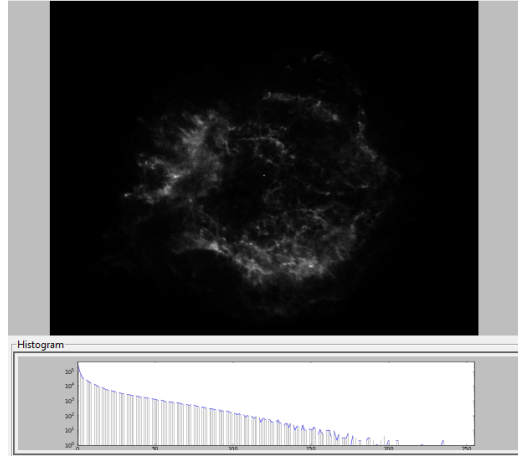


Figura 10: Ajuste de rango dinámico lineal

##### 2. Raíz cuadrada

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz, realiza la siguiente transformación (Ver figura 11):

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

##### 3. Logaritmo

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz, realiza la siguiente transformación (Ver figura 12):

$$Factor = \log_{10}(M - m) \quad Pixel[x, y] = \frac{\log_{10}(Pixel[x, y])}{Factor}$$

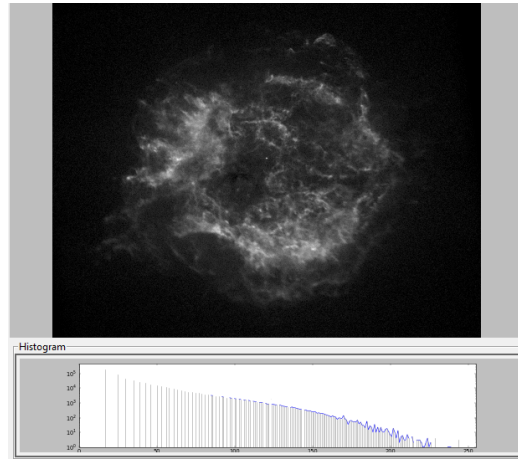


Figura 11: Ajuste de rango dinámico Raíz Cuadrada

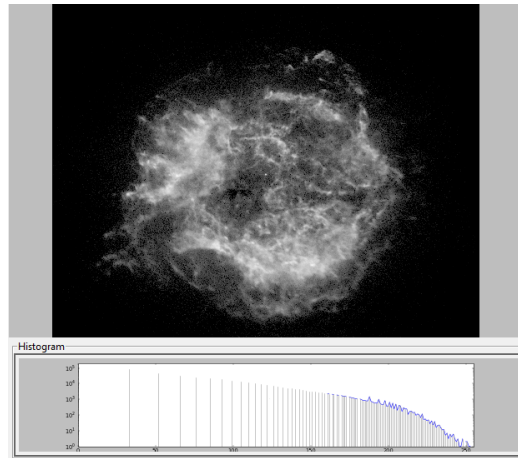


Figura 12: Ajuste de rango dinámico logarítmico

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 13):

$$Factor = \frac{1}{(M-m)^n}$$

$$Pixel[x, y] = (Pixel[x, y] - m)^n * Factor$$

5. Arcoseno Hiperbólico

Conocidos el máximo  $M$  y el mínimo  $m$  valor en la matriz y dado

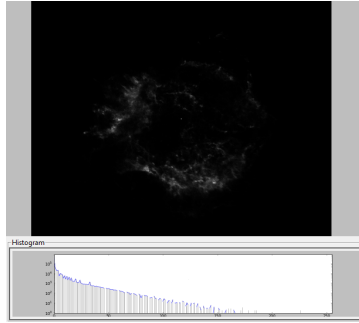


Figura 13: Ajuste de rango dinámico potencia con valor 1.5

un factor de no linealidad  $nL$ , realiza la siguiente transformación (Ver figuras 14 para  $nL=1.5$  y 15 para  $nL=5$ ):

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

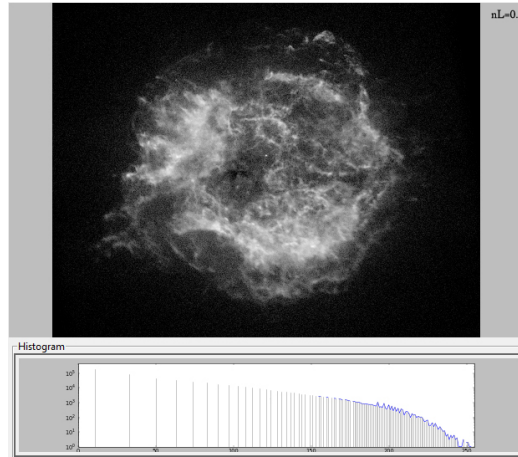


Figura 14: Ajuste de rango dinámico ArcoSeno hiperbólico con valor 1.5

#### 6. Ajuste en base al histograma

La ecualización del histograma de una imagen es una transformación que pretende obtener para una imagen un histograma con una distribución uniforme. Es decir, que exista el mismo número de pixels para cada nivel de gris del histograma de una imagen monocroma (Ver figura 16).

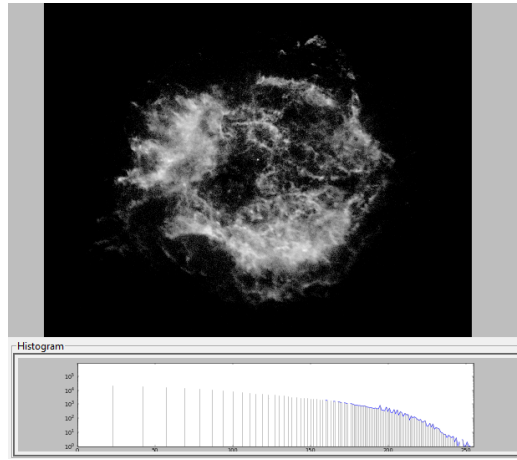


Figura 15: Ajuste de rango dinámico ArcoSeno hiperbólico con valor 1.5

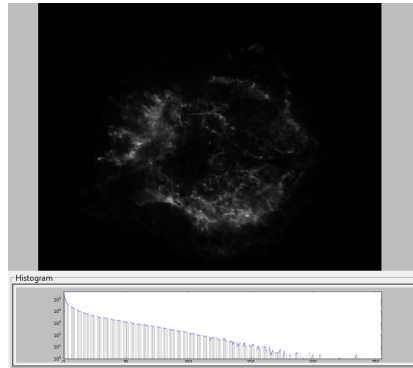


Figura 16: Ajuste de rango dinámico por ecualización de histograma

#### 4.3.2. Búsqueda de puntos de interés

La función `getObjectList` presente en `cvSpace.py` hace uso de la técnica de detección de blobs de `openCV` para localizar puntos de interés.

La búsqueda mediante blobs se centra en la variación de la intensidad de cada pixel con su vecindad, el proceso, en detalle, realiza las siguientes operaciones:

1. Convierte la imagen a una imagen binaria.
2. Extrae las componentes conectadas en base a un cálculo de contornos.
3. En caso de blobs muy cercanos entre si, los agrupa en un único blob,

esta proceso nos es de especial interés para la agrupación de estrellas que conforman una galaxia.

4. Finalmente para cada grupo de blobs nos devuelve tanto su centro como su área.

#### 4.3.3. 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.

#### 4.3.4. 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, es de un tamaño de 5.

Un kernel es una matriz que se aplica a cada pixel de la imagen, expresando este procedimiento en una ecuación tenemos:

$$K(i, j) = 1 \forall x, y \in (0, 5)$$

Nota: En nuestro caso  $K(i, j) = 1$  ya que es la matriz unidad.

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

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

#### 4.3.5. Determinación de forma en las galaxias

### 4.4. Línea de procesamiento de imágenes

#### 4.4.1. Pre-procesado base

El siguiente esquema detalla el proceso que sigue cada imagen, a este proceso lo llamaremos pre-procesado base. (Ver figura 17)

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

El resultado de este pre-procesado da como resultado 2 matrices o imágenes, que denominaremos imagen Halo e Imagen Candidatos.

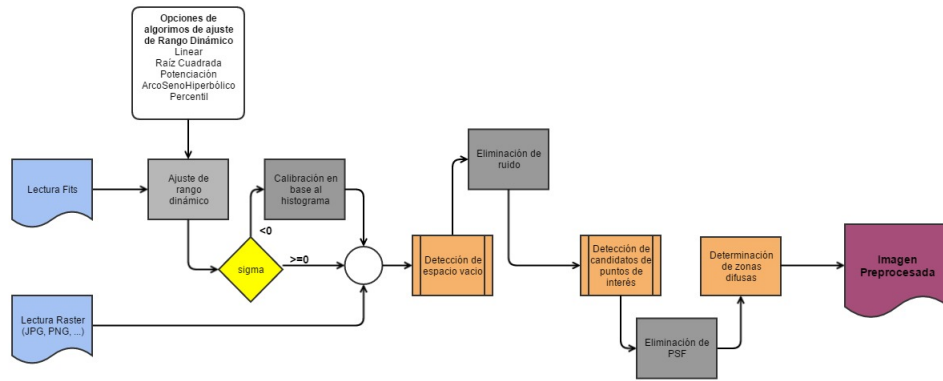


Figura 17: Flujo de pre-procesado

#### 4.4.2. 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 18)

Obtención de coordenadas de puntos de interés: se genera una lista de candidatos partiendo de la imagen Candidatos obtenida tras el pre-procesado.

Para cada coordenada candidata, se valora 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).

### 4.5. Interfaz gráfica

Tanto para el usuario final como para la validación de los algoritmos en tiempo de desarrollo se ha creado una interfaz, esta permite la visualización de la imagen, ajuste de parámetros y obtención en forma visual de los elementos astronómicos detectados así como su clasificación. Igualmente permite el guardado de la imagen resultante y la exportación del proyecto.(Ver figura 19)

En las siguientes líneas se detalla su uso: (Por completar cuando la interfaz este cerrada)

La parte central de la interfaz la ocupa la imagen, es posible aumentar, reducir y trasladar la imagen.

Bajo esta, se encuentra el histograma de la imagen, junto con tres líneas que marcan la segmentación, concretamente la línea negra marca el valor de cielo oscuro, la línea verde marca el halo de galaxias o PSF de estrellas y la línea amarilla delimita el mínimo a partir del cual se considera estrellas. Esta información es la que usa el algoritmo de determinación para categorizar los elementos.

Las tres imágenes situadas bajo el histograma muestran un previo de las zonas de segmentación.

Desde el menú superior, podemos cargar nuevas imágenes

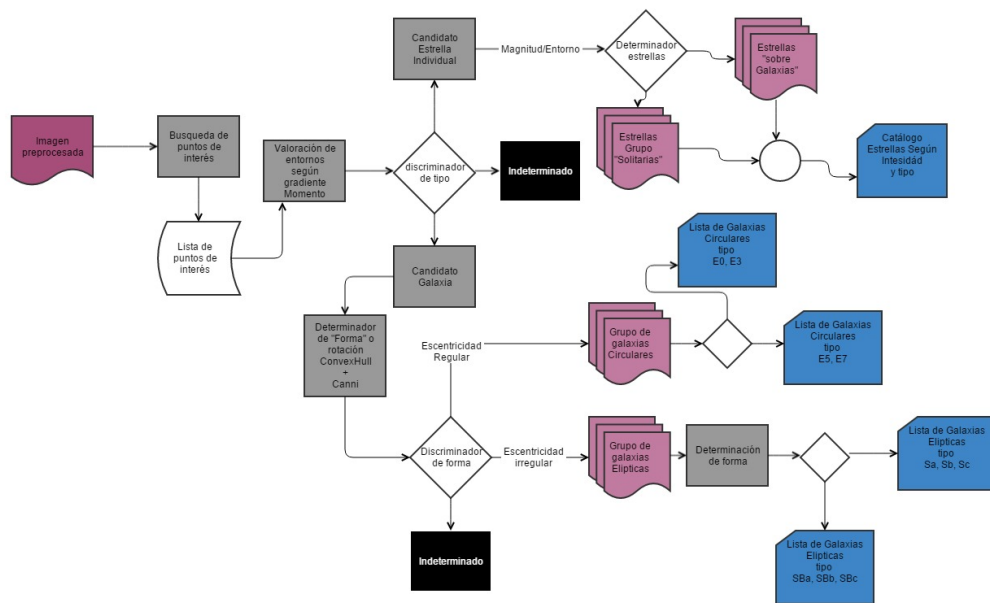


Figura 18: Flujo de clasificación



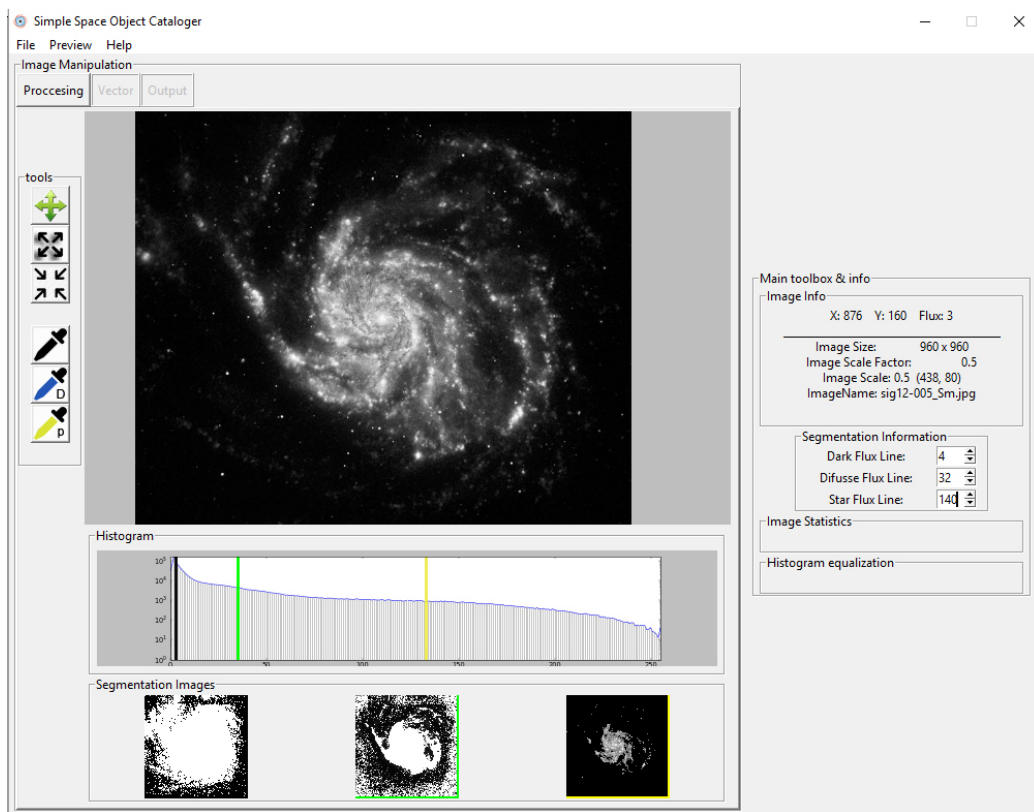


Figura 19: Interfáz gráfica

## 5. Conclusiones

[Por hacer al finalizar el desarrollo]

### 5.1. Cumplimiento de objetivos

### 5.2. Trabajo futuro

El trabajo, aun siendo plenamente funcional presenta aspectos abiertos a mejoras.

- Los ajustes de histograma no funcionan correctamente en todos los casos, sería necesario aumentar el pre-análisis de los datos crudos para detectar cual sería el algoritmo más apropiado.
- Los ajustes heurísticos requieren afinamiento, no se descarta tomar nuevas estrategias.
- Optimizar, el código puede ser mucho más óptimo. 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, en algunos casos se duplican innecesariamente imágenes (thumbnails).

## 6. Agradecimientos

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

A Miguel Angel Valero Espada por darme un enfoque distinto al campo de la programación.

Ana Inés Gomez de Castro por sus enseñanzas en Astronomía.

Especialmente a Rocío por animarme en todo momento y a mi familia por su apoyo.

Sin estos apoyos no hubiese sido posible completar este trabajo.

Este trabajo está dedicado especialmente a mi Yaya Mercedes y a mi Yeyo Modesto, por todos los valores que me supieron dar.

## 7. Referencias bibliográficas