

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“DETECCIÓN DE GALAXIAS OSCURAS UTILIZANDO MÉTODOS DE *DEEP LEARNING*”

JORGE REINALDO LUDUEÑA GIACAMAN

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Juan Ricardo Ñanculef Alegria

Profesor Correferente: Umberto Rescigno

Agosto - 2023

DEDICATORIA

Dedico este trabajo a mi familia y amigos que me apoyaron a lo largo de este viaje y en especial a mi padre que nunca dudo de mí.

AGRADECIMIENTOS

Quiero agradecer al Profesor Ricardo Ñanculef y a Umberto Rescigno por haberme apoyado a lo largo de esta travesía, dando respuesta a mis dudas y ayudándome a arreglar la serie de errores que cometí a lo largo del viaje.

Además, quiero agradecer a mis amigos que conocí en la universidad, quienes me recibieron el primer año y me ayudaron a acostumbrarme a la vida en Santiago y universitaria que me permito llegar hasta este momento.

También agradecer a mis amigos de Curicó que a pesar de la distancia nunca perdimos comunicación y me apoyaban a la distancia.

A mi familia que siempre me apoyaron y se preocuparon por mí. A mis hermanos que me recibieron con los brazos abiertos en Santiago. También a mi gato Pipo que me acompañó en más de una noche de trabajo.

Por último, quiero agradecer a mi madre que me apoyo en todo momento dándome energías y a mi padre que si bien no vera el cierre de mi carrera sé que estaría orgulloso.

RESUMEN

El efecto de lente gravitacional permite observar y estudiar cuerpos no visibles, como los halos de materia alrededor de la parte luminosa de la galaxia. Sin embargo no existe una gran cantidad de casos aceptados en los cuales este efecto ocurre cuando la lente no es visible. Por esto en este trabajo se plantea generar imágenes de *lensing* donde la lente es un agente *oscuro* utilizando *deeplensstronomy* y utilizar estas imágenes para entrenar un modelo de *deep learning*.

Los resultados obtenidos muestran una serie de imágenes realistas y representativas, algunas de las cuales fueron evaluadas por un experto, quien también ha identificado un cierto porcentaje de casos que presentan características menos consistentes con las observaciones. Además se observó que el modelo entrenado fue capaz de diferenciar los efectos de lente *oscuro* con los efectos de lente brillante, sin embargo no tuvo tanto problema de diferenciar los casos positivos de *lensing* y los caso negativos cuando se quitaban los casos brillantes.

Palabras Clave: Efecto de lente, Sérsic, Lente *oscuro*.

ABSTRACT

The gravitational lensing effect allows us to observe and study non-visible objects, as the halos of matter around the luminous part of the galaxy. However, there are not many accepted cases where this effect occurs when the lens is not visible. Therefore, this study proposes generating *lensing* images where the lens is an *dark* agent using *deeplenstronomy* and using these images to train a deep learning model.

The obtained results show a series of realistic and representative images, some of which evaluated by an expert, who has also identified a certain percentage of cases that exhibit characteristics less consistent with the observations. Furthermore, it was observed that the trained model was able to differentiate between the effects of dark and bright lensing systems, but it had less difficulty distinguishing between positive lensing cases and negative cases when the bright cases were removed.

Keywords: *lensing*, Sérsic, *Dark* lense.

GLOSARIO

Aquí se deben colocar las siglas mencionadas en el trabajo y su explicación, por orden alfabético. Por ejemplo:

ANN: Artificial Neural Network. Utilizada cuando se refiere a una red neuronal

CNN: Convolutional Neural Network. Utilizada cuando se refiere a una red neuronal convolucional

e_1 y e_2 : Componentes complejos de elipticidad.

ϵ : Elipticidad. Mide la compresión de un círculo a lo largo de su diámetro para formar una elipse.

HVC: High Velocity Cloud, Nube de alta velocidad

K : Se refiere a la escala cosmológica a cierto *redshift*

Lensing: Se refiere al efecto de la lente

m : magnitud. Es la medida del brillo de una fuente

n_S : Índice de Sérsic. Controla el ángulo de curvatura del perfil de luz.

R_S : Radio de Sérsic. En este trabajo este valor se considera como *half-light radius* el cual mide el radio en el cual la mitad de la luz de una fuente se encuentra contenida.

z : *redshift*. Es la medida física sobre el aumento de la longitud de onda sobre la radiación electromagnética. Se utiliza como medida de distancia en astronomía.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	V
GLOSARIO	VI
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XII
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
CAPÍTULO 2: MARCO CONCEPTUAL	6
2.1 Antecedentes	6
2.1.1 Astronomía	6
2.1.2 Aprendizaje automático	10
2.1.3 Redes Neuronales artificiales	12
2.1.4 Redes Neuronales <i>Feed-Forward</i>	14
2.1.5 Redes neuronales convolucionales (CNN).	18
2.2 Estado del arte	24
2.2.1 Generación de imágenes	24
2.3 Gravitational Lens Finding Challenge	29
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	36
3.1 Generación de imágenes	36
3.1.1 Cálculo de parámetros	36
3.1.2 Estimación de parámetros	38
3.1.3 Cambio de <i>redshift</i>	40
3.1.4 Generación de imágenes y ruido de <i>deeplenstronomy</i>	41

3.1.5	Proceso para las imágenes de fuentes	43
3.2	Modelo de <i>deep learning</i>	45
3.2.1	Conjunto de datos de entrenamiento	45
3.2.2	Entrenamiento y Evaluación	46
CAPÍTULO 4:	RESULTADOS Y ANÁLISIS	48
4.1	Infraestructura y Librerías	48
4.2	Generación de imágenes	49
4.2.1	Obtención, procesamiento de imágenes y ajuste de medidas	49
4.2.2	Estimación y cálculo de parámetros para las fuentes	55
4.2.3	Parámetros de instrumento, lente y ruido	58
4.2.4	Resultado de la simulación de efecto de lensing	60
4.3	Modelo de <i>deep learning</i>	64
4.3.1	Experimento 1	65
4.3.2	Experimento 2	70
4.3.3	Conjunto de datos N°3	75
CAPÍTULO 5:	CONCLUSIONES	81
5.1	Generación de imágenes	81
5.1.1	Estimación del índice de Sérsic	81
5.1.2	Cambio de <i>redshift</i> , ejes y magnitud	82
5.1.3	Imágenes simuladas con <i>deeplenstronomy</i>	82
5.2	Modelo de <i>deep learning</i>	83
5.2.1	Experimento 1	83
5.2.2	Experimento 2	83
5.2.3	Experimento 3	84
5.3	Trabajo futuro	84
ANEXOS		86
5.1	Resumen de unidades	86
5.2	Ejemplo archivo de configuración de <i>deeplenstronomy</i>	86

REFERENCIAS BIBLIOGRÁFICAS	90
--------------------------------------	----

ÍNDICE DE FIGURAS

1	Lente gravitacional en la configuración de Cruz de Einstein, ocurre cuando la fuente es un quasar (el cual tiene una forma de punto) y se encuentra cerca del centro de la lente en proyección, generando 4 imágenes del quasar al rededor de la lente, dándole el nombre a esta extraña configuración.	4
2	Descripción gráfica de una neurona.	14
3	Conexiones en una red <i>Feed-Forward</i>	14
4	Red <i>Feed-Forward</i> completa de ejemplo.	16
5	Ejemplo de convolución 2D.	22
6	Ejemplo de MaxPooling.	24
7	(a) presenta un modelo sin variación de tamaño, mientras (b) presenta una variación de tamaño a la mitad.	32
8	Modelo <i>baseline</i> de <i>Lensfinder</i>	34
9	Muestra de imágenes de ejemplo obtenidas mediante la API de <i>Legacy Survey</i>	50
10	Resultado después de aplicar el método basado en <i>k-means</i> con $k = 2$ para eliminar cualquier cuerpo luminoso que no se encuentre directamente sobre o detrás de la fuente descrita en el catálogo SGA.	51
11	Resultado de aplicar <i>nlabel</i> al resultado de k-means: los píxeles de color verde son los que se desea detectar como pertenecientes a la galaxia de interés, mientras se deben eliminar los negros y rojos, correspondientes a otros objetos.	52
12	Imágenes originales con objetos extra (en su mayoría) y fondo eliminado.	53

13 Efectos de <i>lensing</i> en escala Viridis (ver texto principal para una mayor discusión).	61
14 Ejemplo obtenido de anillo de Einstein.	62
15 Efectos de <i>lensing</i> con mucha luz.	62
16 Efectos de <i>lensing</i> con "puente".	63
17 Imágenes de ejemplo sin <i>lensing</i> en escala viridis.	64
18 Función de pérdida a lo largo de las <i>epochs</i> de entrenamiento para el Experimento 1.	66
19 Matriz de confusión para el Experimento 1.	67
20 Casos de ejemplo en los cuales el modelo confundió efectos de lente <i>oscuro</i> como casos de sin lente en el Experimento 1.	68
21 Casos de ejemplo en los cuales el modelo predijo efecto de lente cuando este no está presenta en la imagen en el Experimento 1.	69
22 Curva ROC obtenida al comparar la predicción de lentes oscuros vs no lente (Experimento 1).	70
23 Función de pérdida a lo largo de las <i>epochs</i> de entrenamiento para el Experimento 2.	71
24 Matriz de confusión para el Experimento 2.	72
25 Casos de ejemplo en los cuales el modelo no pudo detectar el efecto de <i>lensing oscuro</i> en el Experimento 2.	73
26 Casos de ejemplo en los cuales el modelo observa un efecto de <i>lensing oscuro</i> donde no existe en el Experimento 2.	74

27	Curva ROC obtenida al comparar la predicción de lentes oscuros vs no lente y lente brillante (Experimento 2).	75
28	Función de pérdida a lo largo de las <i>epochs</i> de entrenamiento para el Experimento 3.	76
29	Matriz de confusión para el Experimento 3.	77
30	Ejemplo de cada categoría observable en la figura 29. Notar que no existe el caso que se predice lente <i>oscuro</i> cuando el caso es de lente brillante en el Experimento 3.	78
31	Curva ROC para las dos clases de lentes consideradas en el Experimento 3. . . .	79

ÍNDICE DE TABLAS

1	Muestra de datos de ejemplo obtenidos del catálogo SGA 2020.	49
2	Ejemplos de valores presentes por el catálogo SGA.	54
3	Ejemplos de valores transformados del catálogo SGA.	55
4	Resultados de la estimación del índice de Sérsic.	55
5	Resultado de cambio de los valores originales a y b al asignar un nuevo <i>redshift</i> z' para obtener a' y b'	56
6	Resultado del cambio de escala sobre la distancia de luminosidad según un nuevo <i>redshift</i>	57
7	Resultado del cálculo y estimación de los parámetros del perfil de Sérsic para ser utilizados como fuente bajo el efecto de lente.	58
8	Parámetros y valores	59

9	Reporte de clasificación para el Experimento 1.	70
10	Reporte de clasificación para el Experimento 2.	74
11	Reporte de clasificación para el Experimento 3.	79

INTRODUCCIÓN

El efecto de *lensing gravitacional* es uno de los fenómenos más fascinantes predichos por la Teoría de la Relatividad General de Einstein. Este fenómeno ocurre cuando el campo gravitacional de una agrupación de masa, como una galaxia o un agujero negro, actúa como una lente, curvando y distorsionando la luz proveniente de un objeto de fondo. Detectar y observar este efecto tiene importantes aplicaciones en astronomía y astrofísica. Un ejemplo de aplicación ocurrió en 2015, cuando se logró predecir la aparición de una supernova [Kelly *et al.*, 2016] dentro de una galaxia que estaba bajo el efecto de lente de esta última. Cabe destacar que las supernovas son difíciles de predecir.

Debido a que este efecto depende de la masa de los objetos, la teoría sugiere que permitiría detectar grandes agrupaciones de masa *oscura*, es decir, sistemas que no son observables directamente ya que no emiten luz. Más aún, dado que el efecto de *lensing* depende sensiblemente de las propiedades de la distribución de masa, este fenómeno permitiría estimar las características de sistemas oscuros a partir de los efectos que tienen sobre otros cuerpos luminosos. Si bien existe una gran cantidad de sistemas de *lensing* observables, estos casos corresponden a sistemas compuestos por componentes de masa *oscura* y brillante y hasta ahora se conocen pocos casos candidatos donde el sistema que actúa como lente solo contenga masa *oscura* o presente una ausencia casi total de materia bariónica.

Se han desarrollado una serie de herramientas para simular imágenes sintéticas de *lensing*, como las librerías *lenstronomy* [Birrer y Amara, 2018] y *deeplenstronomy* [Morgan *et al.*, 2021], así como *pipelines* para generar imágenes utilizadas en desafíos [Metcalf *et al.*, 2019], con el objetivo de automatizar la detección de este efecto.

Este trabajo explora la posibilidad de detectar automáticamente candidatos a galaxias *oscuras* aprovechando el fenómeno de *lensing gravitacional*. Para ello, se propone el entrenamiento de modelos de aprendizaje profundo (*deeplearning*) que aprendan los patrones de deformación relacionados con estos sistemas y que permitan detectarlos sobre la canti-

dad masiva de datos que generará la próxima generación de telescopios. Dada la ausencia de datasets etiquetados que permitan entrenar estos modelos, se propone utilizar la librería *deeplenstronomy* para generar imágenes sintéticas del efecto de *lensing* en las cuales la componente luminosa de la masa de la galaxia no es visible en la imagen. Hasta donde sabemos, la idea propuesta no ha sido explorado en la literatura.

Este documento se encuentra estructurado en 5 capítulos. En el primero, se presenta la definición del problema y los objetivos a resolver. En el segundo, se presentan los antecedentes necesarios para entender el trabajo, así como distintos trabajos relacionados con la generación de imágenes de *lensing* y su detección utilizando métodos de *deeplearning*.

En el tercer capítulo, se propone una solución mediante un procedimiento para generar imágenes de *lensing oscuro* y la implementación, un modelo de *deeplearning* con el objetivo de entrenarlo utilizando el *dataset* generado. El cuarto capítulo está dedicado a presentar los resultados obtenidos a partir de esta solución propuesta.

Finalmente, el quinto capítulo documenta las conclusiones obtenidas y se discuten posibles trabajos a futuro en esta área de investigación.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

En el universo existen una gran cantidad de estructuras, como galaxias, planetas, estrellas, nubes de gas, entre otros. Lo que se mantiene igual en todas estas es el hecho de contener materia bariónica [Hawkins, 1997], la cual, en palabras simples, se define como la materia que es posible observar.

Desde aquí se define una galaxia *oscura* como aquella que presenta una ausencia casi total de materia bariónica (en particular, ausencia de estrellas), por lo cual no es observable al no presentar casi ninguna emisión de onda electromagnética, lo que las hace prácticamente invisibles y silenciosas. Por lo cual, su existencia sigue siendo teórica, ya que, como presenta [Hawkins *et al.*, 1997], existe una posible forma de detectarlas mediante efectos que tengan en otros objetos que sí pueden ser detectados. Específicamente en este caso, hablamos de una galaxia *oscura* actuando como lente gravitacional, por lo cual tiene un efecto en ondas electromagnéticas no generadas por la misma galaxia, sino por otra fuente luminosa.

Se define un lente gravitacional como el fenómeno que se genera cuando una distribución de masa, ya sea una galaxia masiva, un *cluster* de galaxias u otra estructura con grandes cantidades de masa, refracta la luz proveniente de una fuente de radiación electromagnética lejana como por ejemplo una galaxia. Debido a este fenómeno esta fuente lejana se ve de manera distinta para el observador, de la misma manera que un lente de cristal refracta la imagen para que la persona pueda observar el mundo de manera correcta. La imagen detectada puede tener cualidades especiales que le dan un gran interés para su estudio como por ejemplo, mediante la magnificación de la luz recibida, un lente gravitacional nos permite observar una imagen más clara de la fuente, una imagen con mas detalle o hasta mas de una imagen de la misma fuente. Otro efecto documentado es permitirnos observar fenómenos raros, como una supernova más de una vez en distintos tiempos [Kelly *et al.*, 2016], esto gracias a la dilatación del tiempo ante la gravedad.

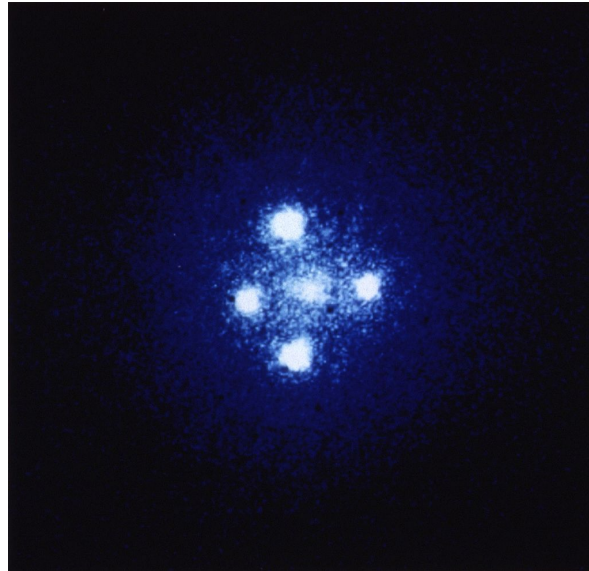


Figura 1: Lente gravitacional en la configuración de Cruz de Einstein, ocurre cuando la fuente es un quasar (el cual tiene una forma de punto) y se encuentra cerca del centro de la lente en proyección, generando 4 imágenes del quasar al rededor de la lente, dándole el nombre a esta extraña configuración.

Fuente: NASA, ESA and STScI

Recientemente, la Universidad de Bologna [Metcalf *et al.*, 2019] (Italia) realizó una competencia en la cual se retaba a los participantes a generar un modelo de inteligencia artificial que permitiera clasificar automáticamente lentes gravitacionales. Este desafío fue lanzado en su primera versión entre 2016 y 2017, con una nueva versión entre 2019 y 2020. Sin embargo, tanto los modelos como los datos utilizados para esta competencia fueron realizados considerando sólo el caso de fenómenos de lentes gravitacionales generados por galaxias normales, es decir, aquellas que poseen un gran número de estrellas y son fácilmente detectables. La pregunta que nace de esto es ¿Es posible generar un modelo que sea capaz de clasificar un lente gravitacional, pero de una galaxia *oscura*?

Considerando que los principios que justifican la aparición de fenómenos de *lensing* valen perfectamente para el caso de galaxias *oscuras*, este proyecto propone extender la iniciativa de Bologna [Metcalf *et al.*, 2019] en una dirección significativamente menos explorada en la

literatura: el descubrimiento automático de este peculiar tipo de galaxias. Para esta tarea se propone el uso de métodos de *deep learning*, la familia de modelos que obtuvo los mejores resultados en el caso de galaxias brillantes ¹.

Se han presentado casos en los cuales lentes gravitacionales se encuentran presentes donde no es posible observar el lente en la imagen. Hawkins [Hawkins, 1997] presenta un caso en el cual se pueden observar dos quasars que presentan cualidades muy parecidas, en donde se podría estar observando un efecto de lensing. Sin embargo comenta que no se puede observar algún otro cuerpo capaz de cumplir el rol de lente. Con esto plantea la opción que simplemente sean dos quasars ligados, pero que por sus similares características y velocidades es difícil aceptar esta opción. O la opción que más acepta, es que el objeto que actúa como lente es una galaxia *oscura* o un halo de materia *oscura*.

Objetivo General

Comprobar la capacidad de reutilización de modelos neuronales entrenados para la clasificación de lentes gravitacionales brillantes, sobre datos de lentes gravitacionales *oscuros*.

Objetivos Específicos

- Construir una serie de imágenes de galaxias *oscuras* usando métodos de simulación.
- Generar un *dataset* correspondiente a imágenes de lentes gravitacionales *oscuros*, imágenes de galaxias sin efecto de lente.
- Seleccionar un modelo con buen rendimiento para la detección de lentes gravitacionales brillantes y entrenarlo sobre el *dataset* generado.
- Documentar y exponer los resultados de la investigación.

¹En este trabajo, se usará el término “galaxia brillante” para referirse a galaxias que emite radiación detectable.

CAPÍTULO 2

MARCO CONCEPTUAL

En este capítulo se plantean antecedentes necesarios para entender los procesos a realizar en este trabajo. Además, se comenta sobre trabajos relacionados con respecto a *lensing*, generación de imágenes sintéticas de este fenómeno y su detección.

2.1. Antecedentes

2.1.1. Astronomía

Galaxias oscuras

Se puede definir de varias maneras una galaxia *oscura*, por ejemplo Hawkins [Hawkins, 1997] las define como galaxias que “fallaron” en crear estrellas, por lo cual no son detectables. Esto puede ocurrir, ya que no contienen materia bariónica o más posiblemente las condiciones para la formación extensiva de estrellas no está presente. Trentham [Trentham *et al.*, 2001] plantea una idea similar donde habla sobre los halos de galaxias *oscuras* pequeñas, las cuales son incapaces de crear estrellas debido a su poca masa, pero existiendo un gran número de estas. Mientras tanto que Jackson [Jackson *et al.*, 1998] las plantea como galaxias *oscuras* que son difíciles de detectar debido a que su ratio masa-luz es muy alto. Marino [Marino *et al.*, 2018] define una descripción parecida en la cual las galaxias *oscuras* son una fase temprana de la generación de galaxias en la cual estas son ricas en gas, pero incapaz de generar estrellas. En el trabajo de Macciò [Macciò, 2008] se plantea una que la masa visible en los satélites en la vía láctea no es lo predicho por el *Cold Dark Matter*, dentro de las posibles razones de esto se plantean están:

- Hay satélites presentes, pero son *oscuros* debido a que se previene la formación de

estrellas.

- Satélites se destruyen debido a la materia oscura que interactúa consigo misma.
- La formación de satélites es impedida por cambiar el espectro de potencia a algo similar a la materia oscura caliente con significativamente menos potencia en las escalas de masa relevante.

Finalmente, Jimenez [Jimenez *et al.*, 1997] plantea que una galaxia oscura es formada con un disco de baja densidad en un halo oscuro con un parámetro de giro alto. Estas pueden tener una densidad de superficie bajo el valor crítico de Toomre para desarrollar inestabilidades. Además, se espera que la generación de estrellas tenga un bajo ratio en estas condiciones.

Para este trabajo, en términos prácticos una galaxia *oscura* es un sistema galáctico en el cual no se detecta o es muy difícil detectar radiación electromagnética, por lo cual no se puede observar directamente. Estos cuerpos pueden estar compuestos solo de materia oscura, de manera que no emiten ningún tipo de radiación (desde ondas de radio hasta rayos gamma). También pueden estar compuestos de una combinación de materia oscura y materia común, como gases, pero sin estrellas que emitan luz. Finalmente, estos cuerpos podrían emitir radiación, pero en rangos de luminosidad que no son detectables por los instrumentos astronómicos que existen en la actualidad.

La búsqueda de galaxias *oscuras* es una empresa en curso desde hace muchos años, pero ninguno de los candidatos encontrados hasta la fecha parece ser ideal. Por ejemplo, el cúmulo sur-occidental (SW) de HI 1225+01 y VIRGOHI21 en el cúmulo de Virgo son algunos de los mejores candidatos a galaxias *oscuras* en la actualidad [Xu *et al.*, 2023]. Sin embargo, ambos candidatos están ubicados en las colas de marea de sus respectivos sistemas, por lo que podrían considerarse posibles restos de marea (tidal debris) [Duc y Bournaud, 2008, Xu *et al.*, 2023]. Otro ejemplo es la nube HVC 127-41-330 [Simon, 2005]. A diferencia de otras HVC (High Velocity Clouds o nubes de alta velocidad) posee una componente rotatoria y un 80 % de la masa es *oscura*.

Como plantea la ESO [Cantalupo *et al.*, 2013], se han realizado variados avances en la forma de detectar, aumentando la sensibilidad de las herramientas utilizadas para observar el universo. Aun así, el número de candidatos a galaxias *oscuras* sigue siendo muy pequeño [Xu *et al.*, 2023]. Esto es relevante por varios motivos. Por ejemplo, como las galaxias *oscuras* ricas en gas podrían reflejar la etapa más temprana en la formación de galaxias, su descubrimiento nos permitiría entender mejor el proceso de formación y evolución del universo. Por otro lado, se ha postulado que la existencia de galaxias *oscuras* podría ofrecer una forma de explicar el bajo número de galaxias ultra-débiles observadas en la actualidad, número que no es claramente consistente con las predicciones del modelo cosmológico estándar (Λ -CDM, Lambda Cold Dark Matter) [Tollerud *et al.*, 2008].

Este proyecto explora la posibilidad de detectar automáticamente candidatos a galaxias *oscuras* aprovechando el fenómeno de *lensing gravitacional*. Como se explica más adelante, y acuerdo a la teoría Relatividad General, un sistema masivo (brillante o no) debiese producir desviaciones de la radiación proveniente de cuerpos brillantes ubicados por detrás de él. Identificando los patrones de deformación sobre la imagen de una fuente de fondo que induce un cuerpo masivo, es posible predecir la existencia de un sistema y estimar sus propiedades, aun si no puede ser observado directamente. Hasta donde sabemos, esta idea se ha utilizado junto a técnicas de *deep learning* para estimar las propiedades de galaxias y cúmulos de galaxias que emiten radiación [Metcalf *et al.*, 2019]. Su uso para el caso de galaxias *oscuras* permanece casi inexplorado en la literatura. Este trabajo es una contribución hacia colmar ese vacío de conocimiento, abriendo nuevas perspectivas en el campo de la detección y estudio de galaxias *oscuras*.

Lentes gravitacionales

Lensing gravitacional se refiere al fenómeno en el cual la dirección de la radiación electromagnética, irradiada por una fuente (llamada fuente bajo el efecto de lente), se ve afectada al pasar por la cercanía de una distribución de masa (la cual se llama lente) que se encuentra entre la fuente y el observador, que en estos casos se considera como la tierra. En este efecto, las ondas electromagnéticas siguen la curvatura de espacio-tiempo que generan los

cuerpos masivos, y es por esta curvatura que el *lensing* gravitacional exhibe rasgos similares a los de las lentes convencionales [Rescigno, 2019].

Los efectos visibles se clasifican en tres clases:

- *Strong Gravitational Lensing*: La imagen de la fuente se divide en múltiples imágenes muy distorsionadas y magnificadas.
- *Weak Gravitational Lensing* : La imagen observada de la fuente se encuentra levemente desplazada y distorsionada. Solamente se observa 1 imagen de la fuente.
- *Micro Gravitational Lensing*: La imagen de la fuente se ve levemente desplazada y su brillo aparente se puede haber visto modificado.

El tipo de *lensing* que ocurre depende tanto de la configuración de posiciones en la cual se encuentre la fuente y a la lente con respecto al observador, al igual que la forma y la cantidad de masa que posea el objeto que actúa como lente.

Los lentes gravitacionales permiten observar fuentes luminosas lejanas, gracias a la refracción característica que generan dependiendo de la forma que tenga el lente. El descubrimiento de los lentes gravitacionales ha permitido realizar más experimentos sobre la teoría de Relatividad General [Einstein, 1916] o generar nuevos métodos para calcular los parámetros cosmológicos como la constante de Hubble [Wong *et al.*, 2019].

Perfil de Sérsic

El perfil de Sérsic [Sérsic, 1963] es la parametrización del perfil de luminosidad de manera que se estime una forma elíptica de una galaxia tal que:

$$I(R) = I_e * \exp \left\{ -b_n \left[\left(\frac{R}{R_S} \right)^{\frac{1}{n_S}} - 1 \right] \right\}; b_n = 1,999n_S - 0,327, \quad (1)$$

donde,

- $I(R)$: Intensidad de la luminosidad en el radio R.

- I_e : Brillo superficial a la distancia del radio medio.
- n_S : Índice de Sérsic.
- b_n : Escalar dependiente del índice de Sérsic n .
- R_S : Radio de Sérsic, definido como radio efectivo o radio medio de luz, radio en el cual la mitad de la luz total es emitida.

La mayoría de galaxias siguen la forma de un perfil de Sérsic con un índice en el rango $1/2 < n_S < 10$ [Caon *et al.*, 1993], además de contener otros perfiles como el de Vaucouleurs [de Vaucouleurs, 1948] el cual es un caso particular en el caso de que se tenga un índice de Sérsic igual a 4.

2.1.2. Aprendizaje automático

El aprendizaje automático permite desarrollar algoritmos capaces de realizar una tarea sin la necesidad de ser programados explícitamente para ello. En palabras de Tom Mitchell “se dice que un programa aprende de la experiencia **E** con respecto a una tarea **T** y un rendimiento medido **P**, es que el rendimiento en **T** medido por **P**, mejora con la experiencia **E**” [Mitchell, 1997].

De manera formal, se define una función o transformación entre dos espacios: uno de entrada y otro de salida. El objetivo es resolver el problema mediante esta función, pero se desconoce la función exacta, por lo que se utilizan diversas técnicas para aproximarla. Específicamente, se buscan los parámetros de la función para que pueda transformar la entrada en la salida esperada. Los valores óptimos de esos parámetros se ajustan a partir de datos o ejemplos del problema que se desea resolver, permitiendo predecir una solución para una entrada dada.

Aprendizaje automático no supervisado

El aprendizaje automático no supervisado se enfoca en el caso en que los datos disponibles no son etiquetados, es decir, no se tienen ejemplos explícitos de las entradas y salidas deseadas. Usualmente, se busca descubrir algún tipo de estructura o patrón en los datos, como el aprendizaje de representaciones significativas, agrupar datos parecidos, y/o descubrir la distribución la cual tienden a seguir los datos.

Uno de los métodos de aprendizaje automático no supervisado se conoce como *k-means clustering* [Smith, 2023]. Este método busca categorizar los datos en K grupos de tal forma que a cada punto se le asigne solamente un grupo. La estructura más simple del algoritmo es:

1. Seleccionar K puntos aleatorios.
2. Por cada punto de data calcular la distancia a cada centroide definido en (1) y asignarle el más cercano.
3. Calcular el promedio de cada grupo y asigna dicho valor como nuevo centroide para el grupo respectivo.
4. Repetir el paso (2) y (3) hasta que el cambio de cada centroide por iteración sea no significativo o se llegue a la cantidad predefinida de iteraciones máximas.

Este método de agrupamiento se puede utilizar sobre imágenes para realizar un *clustering* de colores, trabajando cada píxel de la imagen como un punto en un espacio multidimensional y de esta forma se pueden reducir la paleta de colores para obtener una salida que contenga solamente los K colores dominantes dentro de la foto.

Aprendizaje automático supervisado

El aprendizaje automático supervisado es un enfoque dentro del aprendizaje automático en el que se utiliza un conjunto de datos etiquetados para entrenar un modelo, es decir, se tienen varios ejemplos explícitos de las entradas y salidas deseadas. Esto permite que un algoritmo pueda aprender a realizar predicciones precisas y automatizadas para lograr

inferir el resultado de nuevos datos previamente no vistos. En este aprendizaje, el modelo se ajusta de forma típicamente iterativa a los datos, buscando minimizar una función de error planteada sobre la salida. El objetivo de este procedimiento es aprender una función que sea capaz mapear las entradas a las salidas de manera efectiva.

2.1.3. Redes Neuronales artificiales

Las redes neuronales artificiales (ANN, por sus siglas en inglés) son un método de aprendizaje supervisado que se basa en la estructura y funcionamiento del cerebro humano, con el objetivo de replicar el procesamiento de información que ocurre en las neuronas biológicas. En lugar de utilizar un conjunto localizado de reglas de decisión o un modelo estadístico tradicional, las ANN utilizan un grafo masivo de neuronas artificiales que procesan la información y se adaptan en paralelo durante el entrenamiento [Goodfellow *et al.*, 2016]. Durante este proceso, los pesos y los sesgos de las conexiones entre las neuronas se ajustan para minimizar el error en la predicción de la salida deseada.

Neurona

Cada neurona de una red neuronal es la unidad básica encargada de realizar cálculos según las entradas que reciba y produce una salida en función de estas. Cada entrada tiene un peso asociado, y la neurona aplica una función de activación a una suma ponderada de las entradas y los pesos. Esta salida luego se transmite a las neuronas siguientes en la conexión. En esta estructura los pesos de cada neurona se ajustan para minimizar el error de la predicción basado en la salida deseada.

$$f(x) = \sigma \left(\sum_i^n w_i \cdot x_i + b \right), \quad (2)$$

donde,

- σ : Función de activación.
- n : Dimensión de los vectores de pesos y de entrada.

- w_i : i-ésimo valor del vector de pesos.
- x_i : i-ésimo valor del vector de entrada.
- b : factor de sesgo.

La función de activación se decide para cada capa de la red neuronal. Se trata de una función no-lineal que determina la salida que enviará la neurona. Algunos ejemplos son:

- **Sigmoidal**: Función no-lineal que comprime los valores de entrada entre 0 y 1. Es útil en problemas de clasificación binaria debido a los valores que esta entrega:

$$Sigmoidal(x) = \frac{1}{1 + e^{(-x)}}. \quad (3)$$

- **ReLU (*Rectified Linear Unit*)**: Función lineal por partes que trunca los valores de salida negativos a 0 y mantiene los valores positivos iguales. Utilizada principalmente en capas intermedias de una red y ha demostrado ser efectiva para entrenar redes profundas:

$$ReLU(x) = \max(0, x). \quad (4)$$

- **eLU**: Función similar a ReLU en los valores positivos presentando la función lineal de identidad, mientras aplica una exponencial no-lineal a los valores positivos. Permite acercar los valores negativos a cero como *batch normalization* pero con una menor complejidad computacional:

$$eLU(x) = \begin{cases} x, & \text{Si } x > 0. \\ \alpha \cdot (e^x - 1), & x \leq 0 \end{cases} \quad (5)$$

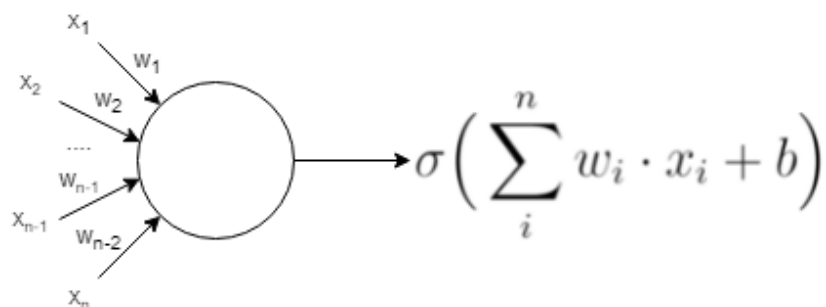


Figura 2: Descripción gráfica de una neurona.

2.1.4. Redes Neuronales *Feed-Forward*

Se denomina red neuronal *Feed-Forward* a aquellas cuyo grafo de computación no forma ciclos, generando un flujo de datos en una única dirección. Cada capa toma la información de entrada de la capa anterior y entrega la información procesada para enviarla a la capa siguiente. Un ejemplo de esta interacción se puede ver en la figura 3.

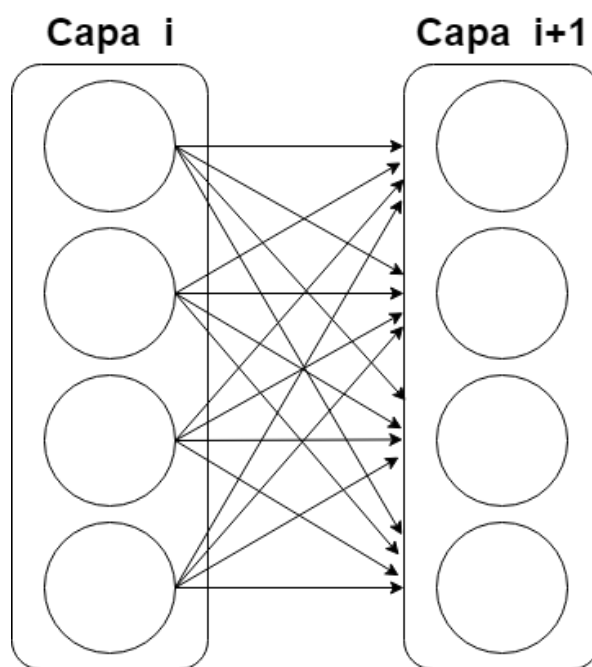


Figura 3: Conexiones en una red *Feed-Forward*.

Las capas de una red *Feed-Forward* se pueden definir en tres grandes categorías:

- **Capa de Entrada:** Es aquella encargada de recibir los datos de entrada originales. A diferencia de las demás, los datos que reciben no están procesados por el modelo, pero sí pueden estar pre-procesados manualmente.
- **Capas oscuras:** Son las que siguen a la capa de entrada y, por lo tanto, donde se realiza la mayor cantidad de procesamiento de información. Estas capas suelen cambiar la dimensión de representación de los datos. La cantidad de capas de este tipo que contenga la red es a lo que determina la profundidad del modelo.
- **Capa de Salida:** Es la última capa de la red, aquella que entrega el resultado final del procesamiento (predicción o solución de la tarea). Se suele utilizar una función de activación especializada en esta enfocada en el formato de salida esperado. Por ejemplo, se suele emplear una función sigmoideal con salidas en $[0, 1]$ para aproximar una predicción que sabemos debe ser binaria, es decir, que toma valores en $\{0, 1\}$.

La profundidad de una red *Feed-Forward* está determinada por la cantidad de capas ocultas. A medida que se aumenta la profundidad de la red, se incrementa la complejidad y capacidad de la red para resolver tareas complejas. La capacidad de la red para aprender y representar patrones complejos depende de la cantidad de capas y neuronas que tenga, lo que le permite a la red aprender características cada vez más abstractas y sofisticadas. Sin embargo, una red demasiado profunda puede llevar a problemas de sobre-ajuste (*overfitting*), en el que la red se adapta demasiado bien a los datos de entrenamiento y no generaliza bien a nuevos datos. Además, una mayor complejidad trae consigo una serie de otros peligros al momento de entrenar la red. Por ejemplo, puede ocurrir el desvanecimiento de gradiente. Un ejemplo de una red *feed-foward* se observa en la figura 4.

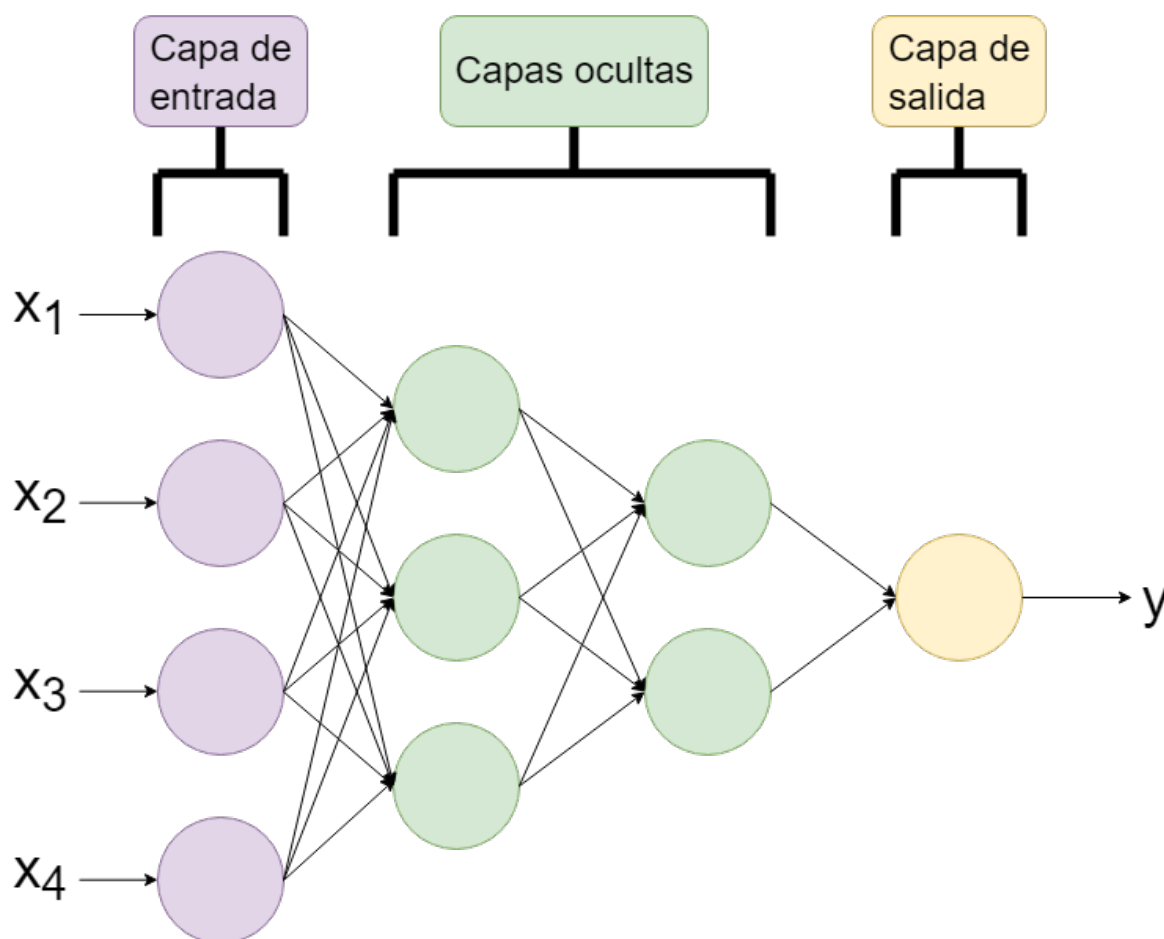


Figura 4: Red *Feed-Forward* completa de ejemplo.

Entrenamiento

Como se plantea en 2.1.2, para el aprendizaje supervisado se habló sobre la necesidad de ajustar el modelo en función de los datos de entrada. A este proceso se le conoce como entrenamiento. Lo primero que se debe realizar es seleccionar una función de pérdida, la cual busca comparar el resultado que sale de la red y' con la salida esperada y (usualmente la salida correcta para una entrada determinada). Algunos ejemplos de funciones de pérdida son:

- *Mean Squared Error*: También conocida como error L2, es el cuadrado de la diferencia entre la predicción y el valor real. Enfocada en penalizar los errores grandes por sobre

los pequeños, se define como

$$L(y, y') = (y' - y)^2. \quad (6)$$

- *Binary Cross Entropy*: Función que se utiliza en los casos que los resultados obtenibles son binarios (generalmente representados con 0 y 1):

$$L(y, y') = (y * \log(y') + (1 - y) * \log(1 - y')). \quad (7)$$

Una vez definida nuestra función con la cual se entrenara la red neuronal, el objetivo se vuelve minimizar el error de entrenamiento:

$$\min_W \frac{1}{N} F_{ann}(W) = \frac{1}{N} \sum_{n=0}^N L(y^{(n)}, y'^{(n)}) = \frac{1}{N} \sum_{n=0}^N L(y^{(n)}, \sigma(W_l x_l + b_l)), \quad (8)$$

donde,

- W : Matriz de pesos de la red.
- N : Número de datos.
- F_{ann} : Función de la red neuronal.
- L : Función de pérdida.
- $y^{(n)}$: Valor esperado de salida en el caso n-ésimo.
- $y'^{(n)}$: Valor obtenido por la red en el caso n-ésimo.
- $\sigma(W_l x_l + b_l)$ Función de activación de la última capa de la red.

El algoritmo utilizado para lograr el criterio de optimización más usado es *backpropagation* [Rumelhart D. E. Hinton, 1986], el cual permite resolver la dependencia dentro de la red con respecto a la función de activación, ya que en cada capa (menos la primera) la activación depende por el resultado de la capa que le precede. La solución consiste en aplicar la regla de la cadena a lo largo de las capas para determinar el gradiente de la función de error

respecto de las activaciones de cada capa, comenzando por la última capa hasta llegar a la primera capa. Este proceso normalmente se implementa mediante programación dinámica por razones de eficiencia computacional.

Una vez calculados los gradientes de cada capa, *Backpropagation* utiliza descenso del gradiente para generar un paso de optimización hacia un mejor valor de la función objetivo (usualmente, el error de entrenamiento). Si W_{t-1} denota un parámetro de la red en la iteración t y $\nabla J(W_t)$ el gradiente correspondiente, el algoritmo aplica la siguiente regla de optimización:

$$W_{t-1} = W_t - \nabla J(W_t). \quad (9)$$

A esta ecuación se le puede añadir un controlador al cual se le conoce como optimizador, que permite controlar la magnitud de los cambios que ocurren en cada iteración, ya sea haciéndolos más o menos drásticos dependiendo del caso y optimizador. El optimizador más utilizado en la práctica suele ser Adam (*adaptive moment estimation*) [Kingma y Ba, 2014], el cual se basa en *AdaGrad* y *RMSProp*, obteniendo los beneficios de ambos para poder acelerar la convergencia de la optimización.

2.1.5. Redes neuronales convolucionales (CNN).

Convolución

La convolución es una operación matemática que se realiza entre dos funciones, generalmente en el dominio del tiempo o el espacio, y se utiliza para analizar cómo una de estas funciones afecta a la otra. Es especialmente útil en el procesamiento de señales para estudiar cómo éstas responden a un impulso en el tiempo. En el caso de un espacio con valores reales se obtiene la expresión:

$$s(x) = (f * g)(x) = \int_{-\infty}^{\infty} f(a) \cdot g(x - a) da, \quad (10)$$

donde

- $f(x)$: Señal en un tiempo continuo x .
- $g(x)$: Impulso en un tiempo continuo x .
- $s(x)$: Resultado de la superposición del impulso por sobre la señal.

Sin embargo, al momento de trabajar en computación, los valores con los cuales se trabajan son discretos, por lo cual la definición de la convolución se modifica para sea calculable de forma discreta, obteniéndose la expresión:

$$s(x) = (f * g)(x) = \sum_{a=-\infty}^{\infty} f(a) \cdot g(x - a) . \quad (11)$$

Dentro del mundo de las CNN, f se concederá como input, g como *kernel*. En el caso más específico de imágenes, las cuales se trabajan como arreglos bidimensionales, se le suele aplicar de la misma forma un kernel con las mismas medidas, obteniéndose la expresión

$$S(x, y) = (I * K)(x, y) = \sum_{b=-\infty}^{\infty} \sum_{a=-\infty}^{\infty} I(a, b) \cdot K(x - a, y - b) , \quad (12)$$

donde

- $I(x, y)$: Imagen de entrada.
- $K(x, y)$: Kernel a ponderar.
- $S(x, y)$: Resultado.

Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son especializadas en reconocer patrones locales en problemas donde la relación de posición de los datos es importante, especialmente en tareas visuales o de procesamiento de imágenes. Lo que hace a las CNN especialmente interesantes es la reducción de parámetros necesarios para trabajar con datos de gran tamaño, como fotografías. Por ejemplo, si consideramos una imagen de 256x256 píxeles con tres canales (rojo, verde y azul), en una red *Feed-Forward* se requeriría

asignar una neurona de entrada a cada píxel. Esto significa que en la capa de entrada tendríamos 196,608 neuronas. El problema surge al definir una segunda capa, donde cada neurona tendría una conexión con su propio conjunto de parámetros provenientes de la gran capa inicial.

Además de los problemas relacionados con la cantidad de parámetros, también se enfrenta el desafío de capturar características distintivas en las imágenes. Debido a que una característica puede desplazarse a lo largo de la imagen y seguir perteneciendo a la misma categoría, se necesita abordar este problema. Por ejemplo, si se está trabajando con una red capaz de detectar bicicletas, se esperaría que la bicicleta esté siempre en la misma posición, lo cual dificultaría su detección si aparece en otra posición, como por ejemplo, con las ruedas hacia arriba. Aquí es donde las CNN son capaces de solucionar estos problemas mediante técnicas:

- *Local Connectivity* (Conexiones locales): La idea es limitar el campo receptivo de las neuronas con tal de que se enfoquen solamente en un área de la información o imagen que se les presenta y con esto sea capaz de encontrar los patrones en dicho subconjunto de datos.
- *Parameter Sharing* (Pesos compartidos): La forma de limitar la cantidad de parámetros sin podar excesivamente la conectividad es agrupar las neuronas y forzarlas a compartir. Esto agrega la restricción de optimización que cada grupo debe tener el mismo peso. Al grupo o activación del grupo de unidades se le suele denominar *feature map*.

Estas ideas planteadas presentan el mismo efecto a si se realizara a la operación de convolución 2.1.5.

Capa convolucional

Se denomina capa convolucional a una capa de una red neuronal que aplica convoluciones, obteniendo los beneficios previamente descritos. Más exactamente, capa convolucional aplica una serie de filtros convolucionales a los datos de entrada utilizando diferentes kernels. A esto se le añade el proceso de la función de activación, la cual se aplica elemento por ele-

mento. Aplicando esta definición y considerando un único filtro, la salida de la capa l estaría definida por:

$$y_l = \sigma \left(W_l \circledast y_{l-1} + b_l \right), \quad (13)$$

donde

- y_l : Vector de salida de la capa l .
- σ : Función de activación.
- W_l : Filtro de la capa l .
- b_l : Vector de sesgo de la capa l .

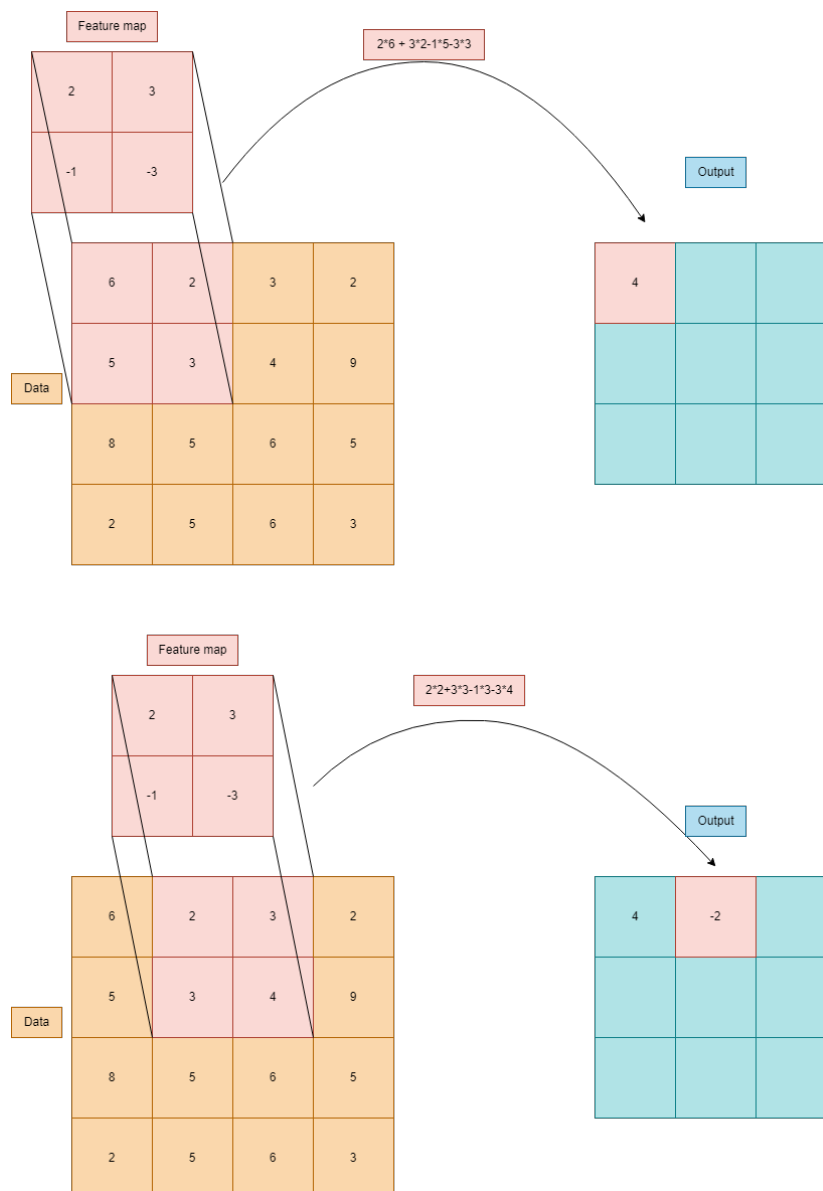


Figura 5: Ejemplo de convolución 2D.

En el ejemplo 5 se puede observar que se aplica un filtro de 2x2 sobre un volumen de 4x4 que produce un resultado de 3x3. Dependiendo de las variables que se eligen, las medidas de salida pueden cambiar. Por ejemplo, si se añaden 5 filtros a la capa, la salida pasaría a ser un volumen 3x3x6. Las variables que afectan esto son:

- **Stride:** Se refiere al movimiento que realiza el filtro a través de los datos. En 5 se pre-

senta un stride de 1.

- **Número filtros:** Cantidad de filtros paralelos que utiliza cada capa, puede ser 1 o más.
- **Tamaño de Filtro:** El tamaño de los filtros suele ser definido de forma cuadrada. Sin embargo, es posible utilizar un filtro con lados distintos.
- **Padding:** Se refiere a añadir datos nulos a los bordes de los datos, generalmente 0, de tal forma que la red sea capaz de identificar patrones en los bordes de los datos de mejor forma.

Pooling

Las diferentes capas convolucionales de una CNN permiten extraer suficientes características visuales con las cuales alimentar red *Feed-forward* para obtener predicciones. Sin embargo, como las capas convolucionales estándar no reducen sustancialmente la dimensionalidad de los tensores de datos que procesan, la cantidad de parámetros de este último modelo podría ser excesivamente grande. Para solucionar este problema, se añade una capa conocida como *pooling*, la cual tiene el objetivo explícito de reducir la dimensionalidad del volumen de datos. Para esta capa se define la forma del campo receptivo y el stride, de la misma forma que se realiza en una convolución. Sin embargo, el stride debe ser definido específicamente para disminuir la dimensionalidad y suele elegirse 2. Además, la dimensión de esta capa se define de forma consistente con las capas convolucionales, es decir, si se está trabajando con una capa en una dimensión, la capa de *pooling* también se encontrará en una dimensión, y así a medida que se aumenta de dimensionalidad.

Un factor diferenciador importante de capas de *pooling* ocurre al momento de determinar la operación que se aplica sobre los datos de entrada. A diferencia de una capa convolucional, las capas de *pooling* usualmente reemplazan los valores observados en el campo receptivo por un único valor. Este valor suele ser elegido de manera simple, ya sea con la media (*Average Pooling*) o el valor mayor (*Max Pooling*). Esta última operación permite a la red ser robusta frente a los cambios de posición de la imagen, como se comentó con el ejemplo de

la bicicleta 2.1.5.

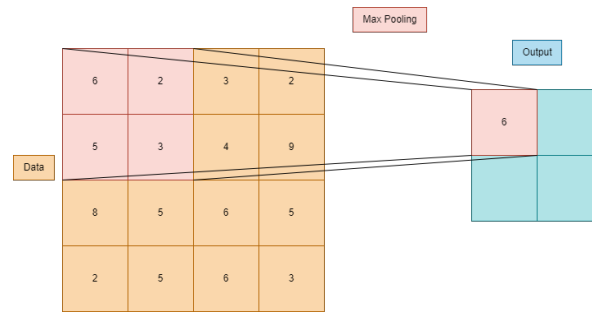


Figura 6: Ejemplo de MaxPooling.

2.2. Estado del arte

2.2.1. Generación de imágenes

Legacy Survey Siena Galaxy Atlas 2020

Legacy Survey presenta un atlas con 383620 galaxias, de longitud de onda múltiple, seleccionadas basándose en su diámetro angular aparente. En este atlas, presentado en forma de catálogo, se incluyen las coordenadas precisas de los astros en cuestión, valores con los cuales se pueden obtener imágenes centradas utilizando la API de *Sky viewer*. Además de la posición de cada sistema, este catálogo es de interés debido a que presenta los valores de *Position Angle (PA)*, *redshift* (longitud de onda que se desplaza hacia el rojo), radio mayor, y radio menor.

Lenstronomy

Lenstronomy [Birrer y Amara, 2018] es un paquete avanzado de *Python* diseñado para el modelado y análisis del fenómeno de lentes gravitacionales. Este destaca por ser una herramienta que se puede considerar estado del arte en el área, debido a su amplio conjunto de funcionalidades y su habilidad de controlar una serie variada de sistemas de lentes gravitacionales. Permite simular sistemas con una o varias lentes, de forma que estas puedan

ser estudiadas por separado. Además, cubre variados aspectos sobre las configuraciones de lentes, permitiendo la simulación de fenómenos de *weak lensing* como de *strong lensing*. Esto se logra mediante un funcionamiento de *framework* modular, en el cual se presentan una variedad tanto de modelos de lentes gravitacionales, como modelos de fuentes que se ven afectadas por el efecto de la lente.

Deeplenstronomy

Deeplenstronomy [Morgan *et al.*, 2021] es un paquete de *Python* afiliado a *Lenstronomy* que engloba sus funcionalidades con el objetivo de simplificar la generación masiva de imágenes y datos de *strong lensing* para su uso en *deep learning*.

Esto se logra mediante un archivo en formato *YAML* que contiene los parámetros de cada objeto en el sistema, así como diferentes configuraciones. Además, permite utilizar distribuciones predefinidas o definir distribuciones propias en caso de ser necesario. Estas variaciones se pueden aplicar a cualquier parámetro presente en el archivo de configuración para agregar variedad al conjunto de datos generado, lo cual es óptimo para el entrenamiento de modelos como redes neuronales.

El archivo en formato *YAML* necesario para su ejecución consta de varias secciones, con parámetros que deben ser definidos para la generación de casos de *lensing*.

1.- DATASET:

En esta sección del archivo de configuración se definen las propiedades del dataset completo, en vez de las imágenes o sistemas por separado. Las variables a definir son:

- NAME: Nombre del *dataset*.
- SIZE: Cantidad de imágenes a generar.
- OUTDIR: Dirección en caso de que se necesite almacenar el *dataset* en disco.

- SEED: Semilla con la cual se inicializan los valores aleatorios.

2.- COSMOLOGY:

En esta sección del archivo de configuración se definen los parámetros de cosmología con los cuales se realizarán los cálculos. Como valor básico para la ejecución de este se requieren los siguientes parámetros:

- H_0 : Constante de Hubble en un *redshift* tal que $z = 0$.
- Ω_m : Densidad de materia en el universo, definida como la relación entre la densidad de materia del universo y el valor de la densidad de referencia llamada *critical density*.

3.- IMAGE:

En esta sección se especifica la información que permite que las imágenes simuladas tengan un mayor realismo. Concretamente, se especifican las siguientes variables:

- `exposure_time`: Número de segundos usados para capturar la imagen.
- `numPix`: Alto y ancho de las imágenes en píxeles.
- `pixel_scale`: Número de arcosegundos por píxel.
- `psf_type`: Describe el modelo para la función de dispersión de punto (PSF) del instrumento [Liaudat *et al.*, 2023] a utilizar.
- `read_noise`: Desviación estándar del ruido generado por el *read-out* en unidades de electrones.
- `ccd_gain`: El número de foto-electrones necesarios para que haya el incremento de una unidad en el valor de la señal analógica, pasando de su representación analógica a su versión digital correspondiente.

4.- SURVEY:

En esta sección se especifican las propiedades del estudio que recolecta el dataset que se quiere simular, planteando los parámetros:

- BANDS: Filtros ópticos utilizados.
- seeing: La anchura a media altura en arcosegundos del PSF medido a partir de las observaciones.
- magnitude_zero_point: La magnitud a la que se mide un foto-electrón cada segundo.
- sky_brightness: Contaminación lumínica medida de la atmósfera en magnitud por arcosegundo cuadrado.
- num_exposures: El número de exposiciones individuales en cada imagen añadida.

5.- SPECIES:

En esta sección se definen todos los objetos o fuentes de ruido que pueden aparecer en la imagen. La librería acepta tres tipos de especies: *GALAXY*, *POINTSOURCE* y *NOISE*.

La especie más importante debido a tener la mayor cantidad de información es *GALAXY*. Esta tiene dos secciones de perfiles de interés a definir, el perfil de luz o *LIGHT_PROFILE* y el perfil de masa o *MASS_PROFILE*.

LIGHT_PROFILE es el perfil más importante al momento de definir la fuente, ya que este define como se distribuye la luz al rededor de la galaxia y es esta distribución la que será sometida al efecto de lente. Se plantea utilizar un perfil de tipo *SERSIC_ELLIPSE* que contiene los siguientes parámetros:

- magnitude: Medida del brillo relativo de una fuente, que indica la relación de luminosidad entre un objeto celeste y un objeto de referencia (a menudo una estrella de referencia) en el sistema fotométrico utilizado.

- `center_x`: Coordenada del centro de la luz en el eje x en arco-segundos.
- `center_y`: Coordenada del centro de la luz en el eje y en arco-segundos.
- `R_sersic`: Radio medio de luz, o el radio que abarca la mitad de la luz de la galaxia.
- `n_sersic`: Índice de Sérsic.
- `e1`: Primera componente del parámetro complejo de excentricidad e .
- `e2`: Segunda componente del parámetro complejo de excentricidad e .

Por el otro lado, *MASS_PROFILE* es el perfil necesario al momento de definir la lente, ya que esto define como ocurrirá la deformación de la luz. Se propone utilizar un perfil *SIE* con parámetros:

- `sigma_v`: Dispersión de velocidad.
- `center_x`: Coordenada del centro de la luz en el eje x en arco-segundos.
- `center_y`: Coordenada del centro de la luz en el eje y en arco-segundos.
- `e1`: Primera componente del parámetro complejo de excentricidad e .
- `e2`: Segunda componente del parámetro complejo de excentricidad e .

La especie *POINTSOURCE* permite generar una fuente en forma de punto para añadir estrellas en el primer plano. Para estas se define simplemente su magnitud. La última especie que se presenta es *NOISE*, para la cual solamente se encuentra implementado el tipo *POISSON_NOISE*. Este modelo aplica esta distribución de manera independiente en cada píxel de la imagen basándose en una media definida en la configuración.

6.- GEOMETRY:

En esta última sección se define una o más configuraciones en las que se encuentran las especies anteriormente especificadas. Para esto, se definen planos por configuración en los cuales se colocan las especies *GALAXY* y/o *POINT_SOURCE*. Además, a cada plano se le define un *redshift* para definir la distancia en la que se encuentra cada uno.

Un ejemplo de un archivo de configuración se puede encontrar en el anexo.

2.3. Gravitational Lens Finding Challenge

En 2019 se lanzó una competencia llamada *Gravitational Lens Finding Challenge* [Metcalf *et al.*, 2019], la cual tenía como objetivo estudiar la eficiencia de distintos métodos de aprendizaje automático para detectar la presencia de lentes gravitacionales en una imagen. En esta competencia, se presentaron a los participantes dos conjuntos de datos sintéticos², y los participantes podían elegir en cuál de los dos participar de forma independiente. Finalmente, a los participantes se les evaluó midiendo la calidad de las predicciones entregadas con la métrica AUROC.

Generación de datos del Challenge

El primer conjunto de datos se denomina *Space-based* y simula la calidad de las observaciones realizadas por el telescopio Euclid. Para esto, se utilizó una escala de píxeles de 0.1 y una función de dispersión de punto (PSF) Gaussiana.

El segundo conjunto de datos se denomina *Ground-based* y busca simular el estudio KiDS [de Jong *et al.*, 2013], el cual se basa en cuatro bandas (u, g, r e i), aplicando la función de dispersión de punto que se presenta en dicho estudio.

En cada caso, se generó un conjunto de entrenamiento de 20,000 imágenes en cada banda de color, y se indicaba si cada imagen correspondía o no a un efecto de lente. Cada imagen tenía un tamaño de 101×101 píxeles. En total, había un universo de 100,000 imágenes, o

²El término "sintéticos" se refiere a datos simulados.

400,000 en el caso de multibanda. A los participantes se les otorgaban 48 horas para cargar una calificación entre 0 y 1, donde 0 indicaba la menor confianza de que la imagen contenía el efecto de lente, y 1 indicaba la confianza más alta.

Para obtener estos datos, el grupo organizador de la competencia *GLENCO*² definió un *pipeline* para generar los lentes sintéticos, basándose en la simulación Millennium [Boylan-Kolchin *et al.*, 2009] y utilizando una simulación de tipo *N-body* que permitiera obtener halos de Materia oscura.

Luego, el catálogo generado se introdujo en el código de lentes de GLAMER [Meneghetti *et al.*, 2010] [Petkova *et al.*, 2014], el cual es una biblioteca de C++ que permite ajustar el perfil de masa de los halos y estimar el radio de Einstein. También se realizó un aumento de datos mediante la rotación de los halos, con el fin de generar un mayor número de lentes con los cuales crear las imágenes.

Para simular los objetos del fondo (que sufren el efecto de lente), se obtuvieron fuentes del *Hubble Ultra Deep Field (HUDF)* que fueron descompuestas en funciones *shapelet* para eliminar el ruido. A partir de esto, se obtuvieron 9,350 imágenes con su correspondiente redshift y coeficientes de las funciones *shapelet* en las cuatro bandas.

Sobre la generación de imágenes, se selecciona una lente del catálogo generado y se elige aleatoriamente una fuente del catálogo de *shapelet*, sujeta a un límite de magnitud por banda. Se define el *redshift* de una fuente desde el *HUDF* para estos objetos. En caso de que este valor sea menor que el de la lente o tenga una diferencia menor a 0.4, se selecciona otra fuente aleatoria que posea otro *redshift*. Es importante destacar que en estos contextos, se utiliza el *redshift* como una medida de distancia. A continuación, se determina la distancia del centro de la lente a su cáustica, y la fuente se posiciona dentro de 3 veces esta distancia. Se debe mencionar que este valor es arbitrario y está diseñado para producir solamente imágenes claras del efecto de lente fuerte.

Para añadir distractores, es decir, fuentes que no son parte del *lensing* y buscan evitar que los modelos aprendan a contar la cantidad de fuentes en la imagen, se utilizan los paráme-

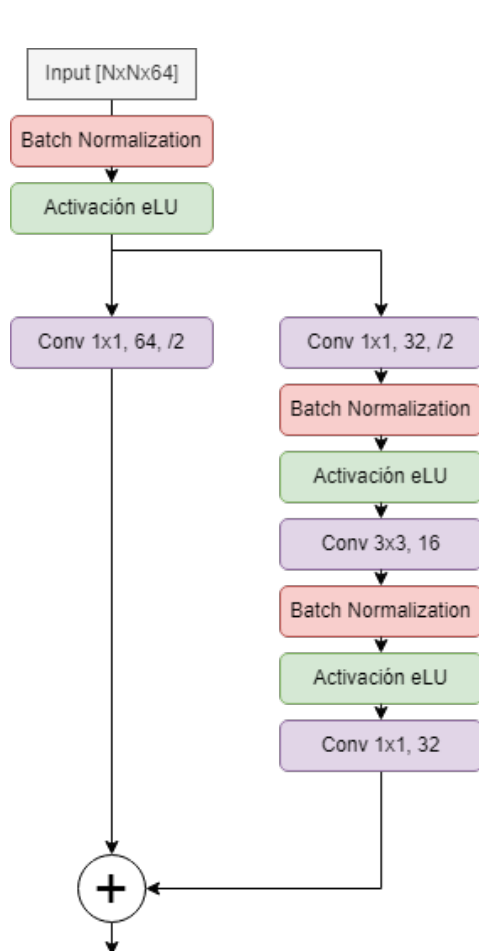
tros entregados por la simulación Millennium que representan las galaxias que habitan los halos de Materia oscura mediante una formación de modelos de galaxias semi-analíticas [Guo *et al.*, 2011]. Los parámetros utilizados incluyen la magnitud total, la relación entre el abultamiento [*bulge*] y el disco, la altura de escala del disco y el radio efectivo del abultamiento [*bulge*].

Ganador de la categoría *Ground-based*

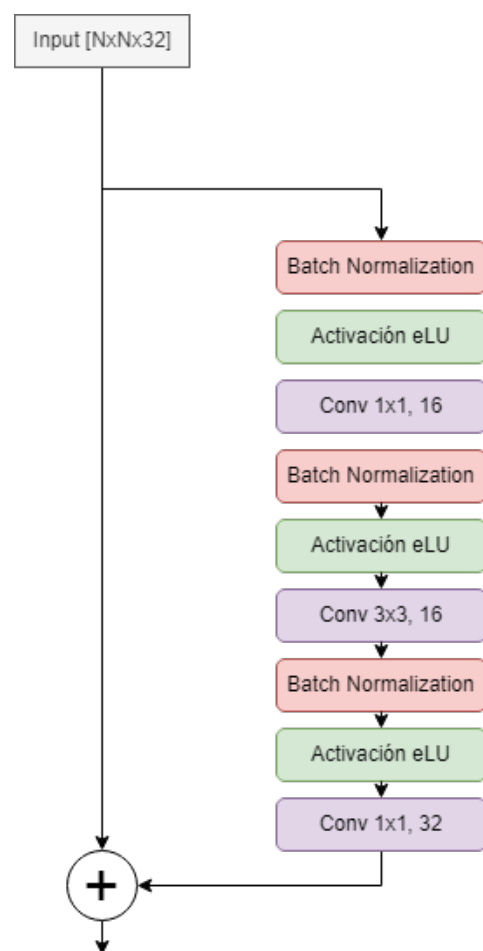
Sobre esta categoría, el modelo ganador fue *CMU DeepLens: deep learning for automatic image-based galaxy-galaxy strong lens finding* [Lanusse *et al.*, 2018]. Esta solución basa su modelo en redes residuales (ResNet) [He *et al.*, 2015], las cuales plantean la idea de usar bloques residuales en los cuales se generan atajos mediante *skip connections* para saltarse un cierto número de capas con el objetivo que la red sea capaz de calcular la representación residual entre la entrada y la salida del bloque. *CMU* utiliza dos estructuras de bloques. La primera mantiene las dimensiones espaciales y de canal de la entrada del bloque. La segunda reduce las dimensiones espaciales a la mitad y aumenta la dimensión de canal al doble. A estos bloques también se añade una pre-activación, es decir, que la función no lineal, característica de cada capa de una red neuronal, se aplica antes de la convolución.

Utilizando bloques con dimensiones de canal entre 32 y 512, se genera la arquitectura de *CMU Deeplens*. Esta arquitectura comienza con un bloque convolucional con activación eLU, seguido de 15 bloques *ResNet*, donde 11 bloques no modifican el tamaño de su entrada y 4 disminuyen la resolución espacial a la mitad. Finalmente, se utiliza un *Average Pooling* conectado a una capa densa de salida sigmoideal y se emplea la función de pérdida *Binary cross-entropy*.

Respecto al preprocesamiento, se realizó una normalización por banda, sustrayendo la media y dividiendo por la desviación estándar σ , teniendo en cuenta todo el conjunto de datos. Además, para restringir dinámicamente el rango de las imágenes de entrada, se eliminaron los valores extremos, truncando los valores más allá de $k\sigma$, con $k = 250$, aunque los autores comentan que si este paso es omitido, no tiene un mayor efecto en los resultados. Además,



(a) Ejemplo de bloque ResNet con entrada de 64



(b) Ejemplo de bloque ResNet con entrada de 32

Figura 7: (a) presenta un modelo sin variación de tamaño, mientras (b) presenta una variación de tamaño a la mitad.

para aumentar la cantidad de datos y entrenar la red ante variaciones de rotación, se realizaron rotaciones aleatorias en el rango de $[-90^\circ, 90^\circ]$, se aplicó reflexión a lo largo de los ejes horizontal y vertical, y se realizaron acercamientos a la imagen/foto en el rango $[0,9, 1]$.

Este modelo obtuvo un AUROC de 0.98, superando a los demás participantes. Sin embargo, es importante destacar que en las pruebas de este modelo se utilizaron datos extra simulados por los investigadores. En el momento de obtener los resultados, la red fue únicamente entrenada con los datos proporcionados por los organizadores.

Ganador de la categoría *Space-based*

En la categoría basada en imágenes espaciales, el trabajo ganador fue *Deep convolutional neural networks as strong gravitational lens detectors* [Schaefer et al., 2018], al cual nombraron *Lensfinder*. Se diseñaron cuatro redes neuronales convolucionales. La primera, denominada *baseline*, se inspira en las arquitecturas que tuvieron un buen rendimiento en la competencia *ImageNet* [Russakovsky et al., 2015]. En esta arquitectura se aplican bloques de dos capas convolucionales seguidas de un *Max Pooling*, y se termina el bloque con una *Batch normalization*. El modelo *baseline* tiene una estructura de tres bloques convolucionales, seguidos de dos capas convolucionales con *Dropout* para evitar el sobreajuste, y finaliza con 4 capas densas, donde la última se presenta como la salida. Para todas las capas se utiliza una versión modificada de *ReLU* definida como:

$$f(x) = \frac{1}{\sqrt{\pi - 1}} (\sqrt{2\pi} \max(0, x) - 1) . \quad (14)$$

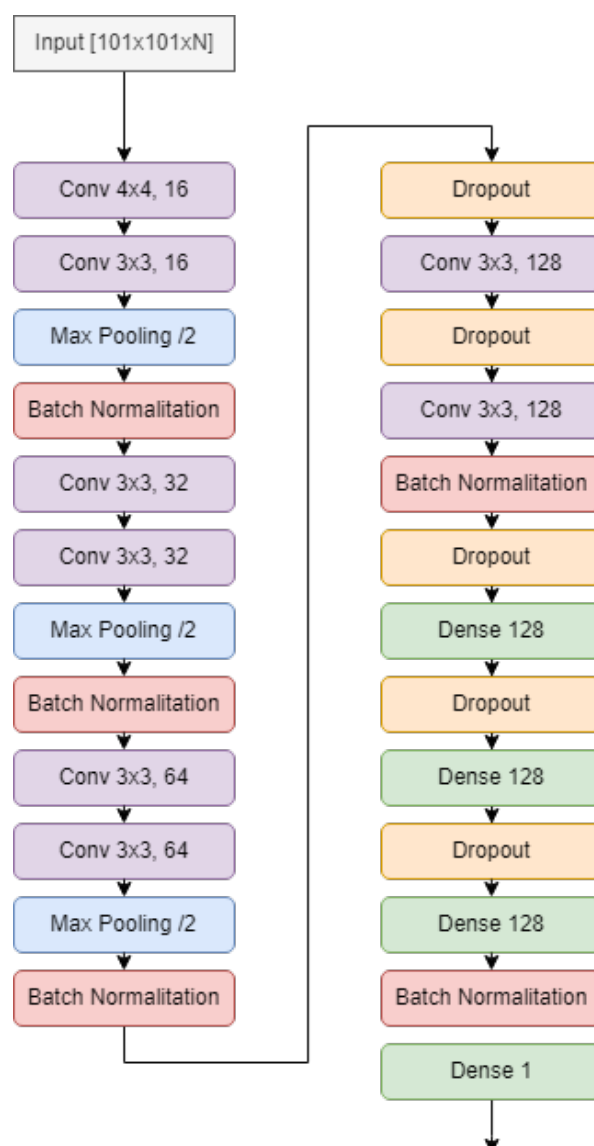


Figura 8: Modelo *baseline* de *Lensfinder*.

En la segunda arquitectura se utiliza la idea de redes residuales en cada bloque convolucional para aumentar la profundidad de la red, añadiendo más capas convolucionales al modelo. Para el segundo modelo, se buscó una mejora en las varianzas del modelo mediante el aumento de datos con la rotación de imágenes y el cambio de tamaño de las mismas. El último modelo busca definir dos caminos dentro de la red, donde uno observa toda la imagen al principio, mientras que el otro solo tiene como entrada una región de 45x45 en el centro de

la imagen, con el objetivo de detectar la lente.

Para las pruebas de entrenamiento se utilizaron 17,000 imágenes para entrenar la red y 3,000 para validar los resultados. Se realizó un entrenamiento de entre 250 y 300 épocas utilizando el algoritmo optimizador ADAM. Como resultado final, no se encontraron grandes diferencias entre todos los modelos presentados, por lo cual se decidió utilizar el *baseline* para enviar los resultados a la competencia, obteniendo un AUROC de 0.92.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

En este capítulo, se presenta una propuesta de solución para la generación de un *dataset* de imágenes correspondiente al efecto de lente gravitacional, donde el objeto que actúa como lente esté compuesto solo de masa oscura. Además, se propone reutilizar un modelo de *deep learning* exitoso en la detección del efecto de lente con lentes brillantes, analizando su desempeño en la nueva tarea.

3.1. Generación de imágenes

Para la generación de imágenes se plantea la utilización de la librería *deeplenstronomy* [Morgan *et al.*, 2021], que permite simular tanto ejemplos del efecto de lente oscura como casos sin dicho fenómeno. Para obtener un resultado más realista al momento de simular escenarios de lente, se plantea utilizar el catálogo de fuentes SGA 2020, el cual contiene una serie de datos e imágenes. Para poder definir las fuentes para generar las imágenes, es necesario poder definir las variables R_S , n_S , m , z , e_1 y e_2 .

3.1.1. Cálculo de parámetros

Dentro del catálogo SGA 2020, se presenta una serie de datos que pueden ser utilizados para definir las fuentes reales. Estos datos son:

- Radio mayor (a).
- Radio menor (b).
- Ángulo de posición, en grados (θ).

- *redshift* (z).
- Magnitud (m).

De estos datos, es posible calcular una serie de parámetros utilizando las definiciones provistas por la documentación de *lenstronomy*. Estos parámetros son:

- Radio medio de luz R_S :

$$R_S = \sqrt{ab}. \quad (15)$$

- Elipticidad ϵ :

$$\epsilon = 1 - \frac{b}{a}. \quad (16)$$

- Ángulo de posición ϕ en radianes

$$\phi = \frac{\theta\pi}{180}, \quad (17)$$

donde θ es el ángulo en grados.

$$e_1 = \frac{\epsilon}{2 - \epsilon} \cos(2\phi) \quad (18)$$

$$e_2 = \frac{\epsilon}{2 - \epsilon} \sin(2\phi) \quad (19)$$

- e_1 : Primera componente del parámetro complejo de excentricidad e .
- e_2 : Segunda componente del parámetro complejo de excentricidad e .
- ϵ : elipticidad.
- ϕ : Ángulo de posición.

3.1.2. Estimación de parámetros

Basándose en los datos entregados por SGA 2020, no se puede obtener un valor de n_s . Para superar esta dificultad, se plantea utilizar un algoritmo de ajuste basado en imágenes para obtener este parámetro. La librería *statmorph* [Rodriguez-Gomez *et al.*, 2018] presenta la función *source_morphology*, la cual realiza una serie de ajustes basándose en una imagen, siendo uno de estos el perfil de Sérsic. Esto último se logra utilizando el método *LevMarLSQ-Fitter* de la librería *astropy* que trabaja refinando una estimación inicial basada en sus ajustes previos.

Para realizar este ajuste, se requiere la imagen de una fuente, y para mejorar los resultados, es conveniente eliminar la mayoría de los objetos que son ajenos a la galaxia a estimar, es decir, cualquier objeto visual presente en la imagen que no corresponda (o este encima) de la galaxia central.

Las imágenes de las galaxias a utilizar se pueden obtener gracias al *Legacy Survey Sky Viewer*, el cual permite observar y obtener imágenes de fuentes si se le entrega las coordenadas del cielo. La posición se presenta en los valores ascensión recta (RA) y declinación (Dec) medida en ángulos. Con estas coordenadas y asignando un tamaño de las imágenes en una escala de píxel en arcosegundo, la API retorna una imagen de la sección del cielo indicada. Dentro del catálogo SGA 2020, de los datos que se presentan de los distintos cuerpos luminosos, se encuentra el centro aproximado en las coordenadas (RA, Dec) los cuales pueden ser utilizados para obtener una imagen de las fuentes en la cual la fuente se encuentra en el centro de la foto.

Detección de objetos con *K-Means* y separación de objetos

Al momento de obtener las imágenes, estas no solo contendrán la galaxia esperada, sino que pueden tener una serie de fuentes de luz al rededor que pueden perturbar la estimación de los parámetros. Para solucionar esto se propone un procesamiento de imágenes basado en *k-means* para la obtención de una imagen solamente donde se encuentre presente solamente

una fuente.

El método consiste en aplicar *k-means* con dos *clústers* a los píxeles de la imagen, de tal forma de que el primer clúster represente el fondo, que típicamente se observa en color negro dominante, y el segundo agrupe cualquier objeto que se aleje del negro, en este caso objetos brillantes. Para esto, la implementación de OpenCV [Bradski, 2000] permite utilizar *k-means* sobre imágenes directamente.

Como resultado se obtendrá una imagen simple que contiene solamente dos colores: el fondo y las entidades brillantes. Para lograr separar la imagen del centro que corresponde a la galaxia objetivo, se utiliza la función *label* que presenta el paquete *ndimage* de *scipy* [Virtanen *et al.*, 2020], a la cual se le entrega una matriz segmentada, en donde los píxeles de lo que se considera el fondo equivalen a 0. La función realiza un escaneo a lo largo de la matriz, asignando una etiqueta inicial basada en *four-connectivity*, que significa que el píxel se considera conectado a sus vecinos del norte y oeste. Este primer escaneo corresponde a:

- Si el píxel equivale a 0, se pasa al siguiente.
- En caso contrario, revisa los vecinos para determinar la etiqueta del píxel.
- Si ambos vecinos tienen la misma etiqueta, se le aplica esta etiqueta al píxel actual, lo mismo se realiza si solamente uno de los vecinos tiene etiqueta y el otro todavía no esta etiquetado.
- Si los vecinos tienen etiquetas distintas, se asigna una al píxel actual y se guarda la equivalencia entre las etiquetas en una estructura de datos de conjunto disjunto.

Después de iterar por todos los píxeles, vuelve a realizar un escaneo a lo largo de la matriz en donde resuelve las etiquetas provisionales basándose en la cadena de equivalencia en la estructura de datos de conjunto disjunto. Finalmente, se tendrá que todas las componentes conexas poseen su etiqueta correspondiente y la función retorna la matriz, pero esta vez el valor de cada píxel será la etiqueta que se le asignó.

Sabiendo que la galaxia se encuentra exactamente en el centro, se elimina cualquier etiqueta distinta a la presente en la posición central de la imagen. Con esto se obtienen todos los píxeles que corresponden a la fuente que se desea estimar los parámetros.

3.1.3. Cambio de *redshift*

Ocurre que el *redshift* que presenta SGA 2020 posee un rango de valores pequeño, por lo cual existe una baja representación de las fuentes luminosas existentes. Para mejorar la cantidad de galaxias representadas, se plantea realizar una modificación de *redshift* a las fuentes de forma aleatoria. Dado que al modificar el *redshift* de una fuente, se está modificando la distancia que esta tiene con la tierra, este procedimiento requiere escalar los valores de tamaño y brillo de la imagen.

Cambio de ejes

Debido a que el *redshift* es una medida relacionada con la distancia de un cuerpo a la tierra, su valor está relacionado con el tamaño aparente de la galaxia. En otras palabras, al modificar el *redshift* z , se debe modificar el tamaño observado de la galaxia. El cambio a aplicar se puede obtener encontrando la relación entre las escalas K y K' (en $\frac{kpc}{arcsec}$) que corresponden a dos valores diferentes de *redshift* z y z' , de acuerdo los parámetros cosmológicos considerados. Wright y Schombert [Wright, 2006] presentan una implementación de este cálculo. En este trabajo, se utilizará la relación de las dos escalas $\frac{K}{K'}$ para transformar los valores originales de los ejes a y b en los valores a' y b' que corresponden al objeto después de modificar su *redshift*. Un resumen de este procedimiento sería:

1. Utilizando el *redshift* original de la galaxia se calcula su escala K .
2. Se multiplica K con el radio menor b y mayor a originales para obtener b_K y a_K respectivamente.
3. Se obtiene un *redshift* nuevo y se calcula su escala obteniendo K' .

4. Finalmente, se obtiene a' y b' al dividir tanto a_K como b_K por K' .

Cambio de magnitud

Al igual que con los ejes, la magnitud presente de una fuente también varía al modificar su *redshift*. Este cambio es en función de la distancia de luminosidad d , la cual corresponde a una medida obtenida basándose en los parámetros cosmológicos y el *redshift* de la misma manera que se calcula la escala. Nuevamente, la calculadora cosmológica de Wright [Wright, 2006] nos presenta este valor para obtener el valor de d .

Una vez obtenida la distancia de luminosidad tanto para el *redshift* antiguo como el nuevo, la transformación de la magnitud se calcula a partir de la siguiente ecuación,

$$m' = m - 5 * \log_{10}\left(\frac{d}{d'}\right), \quad (20)$$

donde

- m' : Magnitud para el nuevo *redshift*.
- m : Magnitud original.
- d : Distancia de luminosidad en el *redshift* original.
- d' : Distancia de luminosidad en el *redshift* nuevo.

3.1.4. Generación de imágenes y ruido de *deeplenstronomy*

Una vez estimados y calculados los parámetros, se obtiene un catálogo de valores del perfil de Sérsic para una serie de fuentes con *redshift* modificados, los cuales pueden ser insertados en los archivos de configuración de *deeplenstronomy* para definir las fuentes del sistema, como se describe a detalle en 3.1.5. Queda así por definir la lente, las fuentes que actúen como distractores, y el ruido del sistema.

Con el objetivo de evitar que el modelo aprenda a contar la cantidad de imágenes presentes de la fuente, debido a que *lensing* fuerte genera múltiples imágenes, se deben añadir a todas las imágenes generadas una serie de fuentes que tengan el único objetivo de perturbar la habilidad del modelo en caso de que intente contar la cantidad de imágenes presente. En la metodología propuesta, estas fuentes distractoras se obtendrán del catálogo ya obtenido, pero se añadirán en distintos *redshift* para aumentar la variación de los datos. Más específicamente, se consideran dos categorías de distractores: una que representa aquel distractor que se encuentra entre la lente y la fuente a recibir el efecto de *lensing* y otra en que el distractor se encuentre más cerca de la tierra que la lente. La aparición de estas fuentes es independiente una de la otra, por lo cual una imagen puede tener entre cero y dos galaxias extra.

Para cada categoría de fuente distractora, se filtra el conjunto de datos original con tal de cumplir los requisitos de *redshift* según su tipo y de esta se obtiene los datos de una fuente, los cuales serán utilizados en el plano definido por su categoría. Debido a limitaciones de la librería, estas fuentes no son capaces de generar efecto de *lensing* por lo cual simplemente actuaran de forma visual.

En el caso de la definición de la lente, se requieren los perfiles de luz y de masa. Para el primero, es de interés que el cuerpo no emita luz visible y finalmente no se logre observar en la imagen, por lo cual al parámetro de magnitud se asigna un valor muy alto y alejado al de las fuentes luminosas reales. Con respecto al perfil de masa, se plantea definir los tres parámetros necesarios e_1 , e_2 y σ_v de manera aleatoria, donde está última seguirá una distribución normal con media $\mu = 250$ y desviación estándar $\sigma = 50$. Esta es una estimación muy aproximada y podría ser mejorable.

Finalmente, para el ruido se utiliza la distribución Poisson, en donde se escala la imagen a un rango de $[0, 255]$ y se le suma a cada píxel de manera independiente los valores obtenidos de la distribución con media 5. El re-escalado se realiza por imagen y una vez que el ruido se ha sumado, la imagen se devuelve a la escala original.

3.1.5. Proceso para las imágenes de fuentes

Finalmente, el proceso completo para la obtención de imágenes se resumiría en los siguientes pasos:

1. Se obtienen las imágenes del catálogo SGA 2020, utilizando el *Sky viewer* y las coordenadas presentes en él catálogo, con el objetivo de que las fuentes se encuentren en el centro de la foto.
2. A las imágenes obtenidas se le elimina el fondo, es decir, cualquier objeto que no sea la galaxia del centro, utilizando el método de *clustering K-means* y el sistema de etiquetados de *scipy label* obteniendo como resultado una imagen con una fuente en su centro y todos los demás píxeles transformados a negro.
3. Las imágenes con fondo negro pasan a ser procesadas de manera visual utilizando la librería *statmorph*, la cual entrega los valores del índice de Sérsic n_S para la definición del perfil de Sérsic.
4. Se obtienen del catálogo SGA 2020 los parámetros de ángulo PA, DA_LEDA, BA_LEDA y MAG_LEDA.
5. DA_LEDA, corresponde al diámetro mayor de la fuente en arcominutos, por lo cual se divide en 2 y se multiplica por 60 para obtener el valor del radio mayor de la fuente original que se denomina a en este trabajo. Posteriormente, este valor se multiplica por BA_LEDA, el cual corresponde al *ratio* entre el radio menor y el radio mayor, para así obtener el radio menor b .
6. MAG_LEDA se transforma de su escala original [Vega mag] a la escala [AB mag] restándole 0.12, obteniendo así m .
7. Se le asigna un nuevo *redshift* a cada fuente de manera uniforme en el rango $[0.5, 4]$.

8. Utilizando el *redshift* entregado por el catálogo SGA y el nuevo *redshift* se obtienen las escalas K y K' correspondientes para cada uno y se calculan los nuevos valores a' y b' .
9. Utilizando el *redshift* entregado por el catálogo SGA y el nuevo *redshift* se obtienen las distancias de luminosidad d y d' con las cuales se transforma el valor original de la magnitud de la fuente en AB mag m en m' .
10. Se calculan los parámetros R_S , e_1 y e_2 en función de a' , b' y PA.
11. Se le asigna un valor aleatorio a la fuente cercana al centro en arcosegundos tal que x e y tengan un rango entre $[-2, 2]$.
12. Teniendo los parámetros calculados R_S , e_1 , e_2 y m' , junto con los parámetros asignados aleatoriamente x , y y *redshift* y el parámetro estimado n_S , se pueden asignar en el archivo de configuración de *deeplenstronomy*, en la sección *SPECIES* dentro del perfil de masa del cuerpo que se utilizara como fuente.
13. Se generan aleatoriamente los parámetros de la lente, tal que tenga un ángulo PA en el rango uniforme $[0, \pi]$, una elipticidad dentro del rango uniforme $[0, 0.7]$ y la dispersión de velocidad con una distribución normal con media $\mu = 250$ y desviación estándar $\sigma = 50$. Los cuales son añadidos a la sección de *SPECIES* como la lente.
14. Se asigna aleatoriamente qué categoría de distractores tendrá. En el caso de que tenga al menos una, se filtra él, catálogo para que se cumpla las reglas de *redshift* dependiendo de la categoría en cuestión, y se añaden los parámetros de la fuente en una especie nueva en la sección *SPECIES*.
15. Con el archivo de configuración definido se utiliza la librería *deeplenstronomy* y se genera las imágenes.
16. Con las imágenes ya generadas se le añade el ruido a la imagen, re-escalándola al rango $[0, 255]$ y sumando de manera independiente una distribución de Poisson con media 5, para posteriormente devolverla a su escala original.

3.2. Modelo de *deep learning*

Se propone reutilizar la arquitectura del modelo *Lensfinder* 8, descrito en la Sección 2.3. Para adaptarlo a la nueva tarea, se plantea modificar la capa salida y reentrenarlo completamente sobre el *dataset* generado.

3.2.1. Conjunto de datos de entrenamiento

Para este proyecto se plantean tres experimentos que difieren en las categorías que identificará el modelo de aprendizaje profundo y los datos empleados para entrenarlo.

1. En el primer experimento, el modelo será entrenado para discriminar casos de *lensing* oscuro de imágenes que no corresponden a este fenómeno. Se tienen, por lo tanto, dos clases diferentes. El modelo será entrenado solo a partir de los datos desarrollados con la metodología descrita en la Sección 3.1. Para prevenir sesgos, se plantea un entrenamiento balanceado entre ambas clases.
2. En el segundo experimento, el modelo también será entrenado para discriminar 2 clases: casos de *lensing* oscuro versus casos que no corresponden a este fenómeno. Sin embargo, el modelo será entrenado combinando datos generados en este trabajo con datos provenientes del *Gravitational Lens Finding Challenge* (GLFC), el cual posee imágenes de *lensing* en la cual la lente es visible e imágenes donde no ocurre dicho efecto. Esto permitirá comprobar que la red es capaz de detectar específicamente efectos de lente oscuro, distinguiéndolos de un caso de *lensing* tradicional. Los ejemplos de la clase *lensing* oscuro serán obtenidos usando la metodología descrita en la Sección 3.1. Los negativos, en cambio, se obtendrán combinando en iguales proporciones: i) datos generados en este trabajo no correspondientes a *lensing* oscuro, ii) datos del GLFC correspondientes a casos de *lensing* tradicional, y iii) datos del GLFC no correspondientes a casos de *lensing* tradicional.

3. En el tercer experimento, el modelo será entrenado para distinguir tres clases: lente oscura, lente tradicional, y no lente. Los ejemplos de entrenamiento correspondientes a la primera categoría se obtendrán usando la metodología descrita en la Sección 3.1. Para obtener ejemplos de la segunda categoría, se utilizarán los casos de lente presentados en el GLFC. Para obtener ejemplos de la tercera categoría, se combinarán en iguales proporciones: i) datos generados en este trabajo no correspondientes a *lensing* oscuro, ii) datos del GLFC no correspondientes a casos de *lensing* tradicional. El objetivo de este experimento será evaluar la habilidad del modelo pueda diferenciar entre efectos de lente brillante, efectos de lente oscura, y casos en donde no hay *lensing*. Para garantizar que el modelo pueda llevar a cabo esta tarea de manera efectiva, es importante asegurar que el conjunto de datos esté balanceado, es decir, que haya una cantidad similar de muestras en cada una de las tres clases.

3.2.2. Entrenamiento y Evaluación

Los detalles de entrenamiento serán los siguientes:

- Función de pérdida: Se plantea la función de pérdida *categorical crossentropy* para entrenar la red en función de varias clases. Esta función queda representada por la expresión:

$$L(y_{\text{true}}, y_{\text{pred}}) = - \sum_i y_{\text{true},i} \cdot \log(y_{\text{pred},i}), \quad (21)$$

donde

- y_{true} : Vector que codifica la categoría real de un dato con un 1 en la posición de la categoría correcta y un 0 en las otras posiciones.
 - y_{pred} : Vector de probabilidades correspondientes a las predicciones del modelo para cada categoría.
- Para cada uno de los experimentos previamente definidos, el total de datos a utilizar se determina calculando primero la cardinalidad n de la categoría con menor cantidad de

datos disponibles y seleccionando igual cantidad de datos n para las demás categorías.

- El entrenamiento se realizará sobre el 80 % de los datos y para pruebas se utilizará el 20 % restante que la red solamente observará para esta evaluación.
- Se utilizarán las siguientes métricas de desempeño:
 - *True Positive Rate (TPR)*, *Recall* o *Sensibilidad*: mide la probabilidad de que el modelo detecte correctamente los casos positivos (lente oscuro) dentro de los datos. El objetivo es que esta métrica sea cercana a 1.
 - *False Negative Rate (FPR)* o *Especificidad*: mide la probabilidad de que el modelo detecte correctamente casos negativos, es decir, que no corresponden a la clase de interés (lentes oscuros).
 - Curva ROC: permite observar la relación entre los *TPR* y los *FPR* al cambiar el umbral de decisión que utiliza el modelo para determinar que un caso de interés es positivo.
 - AUROC: representa el área bajo la curva ROC. Esta métrica global fue utilizada para determinar el modelo ganador en la competencia *Gravitational Lens Finding Challenge* [Metcalf et al., 2019].
 - Exactitud o *Accuracy*: evalúa la cantidad de predicciones correctas.
 - *Precision*: mide la cantidad de verdaderos positivos encontrados por el modelo sobre todos los datos.
 - *F1-score*: utiliza la media armónica entre *Precision* y *Recall* para obtener una única métrica.

CAPÍTULO 4

RESULTADOS Y ANÁLISIS

En este capítulo se presentan los resultados de aplicar la metodología propuesta.

4.1. Infraestructura y Librerías

Para la ejecución se utilizaron los siguientes componentes de hardware:

- **CPU** = AMD Ryzen 5 5600X 6-Core 3.70 GHz.
- **GPU** = NVIDIA GeForce GTX 1060 3 GB.
- **RAM** = 36 GB DDR4 3200 MHz.
- **Disco duro** = TOSHIBA HDWD110.

Para la ejecución se utilizaron las siguientes versiones de librerías:

- **Lenstronomy** 1.11.1. [Morgan *et al.*, 2021]
- **Deeplenstronomy** 0.0.2.3. [Birrer y Amara, 2018]
- **statmoprh** 0.5.0. [Rodriguez-Gomez *et al.*, 2018]
- **Tensorflow** 2.12.1. [Abadi *et al.*, 2015]
- **Keras** 2.12.0. [Chollet, 2015]

4.2. Generación de imágenes

En esta sección se resumen los resultados obtenidos al pre-procesar los parámetros de las fuentes a utilizar como base para simulación y las características de las imágenes finales obtenidas.

4.2.1. Obtención, procesamiento de imágenes y ajuste de medidas

Basado en las posiciones *RA* y *DEC* presentes en el catálogo de fuentes SGA y utilizando la API que pone a disposición *Legacy Survey*, se obtuvieron imágenes centradas de aproximadamente 160,526 fuentes luminosas en una escala de 256x256x3 píxeles con una escala de píxel de 0.27 arcosegundos, en formato PNG. La Tabla 1 y la Figura 9 exponen una muestra de los parámetros e imágenes obtenidas.

Galaxy	RA	Dec
PGC1184971	199.509789	1.114087
IC1069	222.693818	54.411187
PGC1490363	349.178686	15.665165
PGC1321218	125.693330	7.414073
PGC2440228	244.615628	53.406699
PGC2416037	175.051236	52.479692
2MASXJ08334	128.419841	33.606809
PGC1452753	133.699916	14.230645
PGC1542423	152.840725	17.814198

Tabla 1: Muestra de datos de ejemplo obtenidos del catálogo SGA 2020.

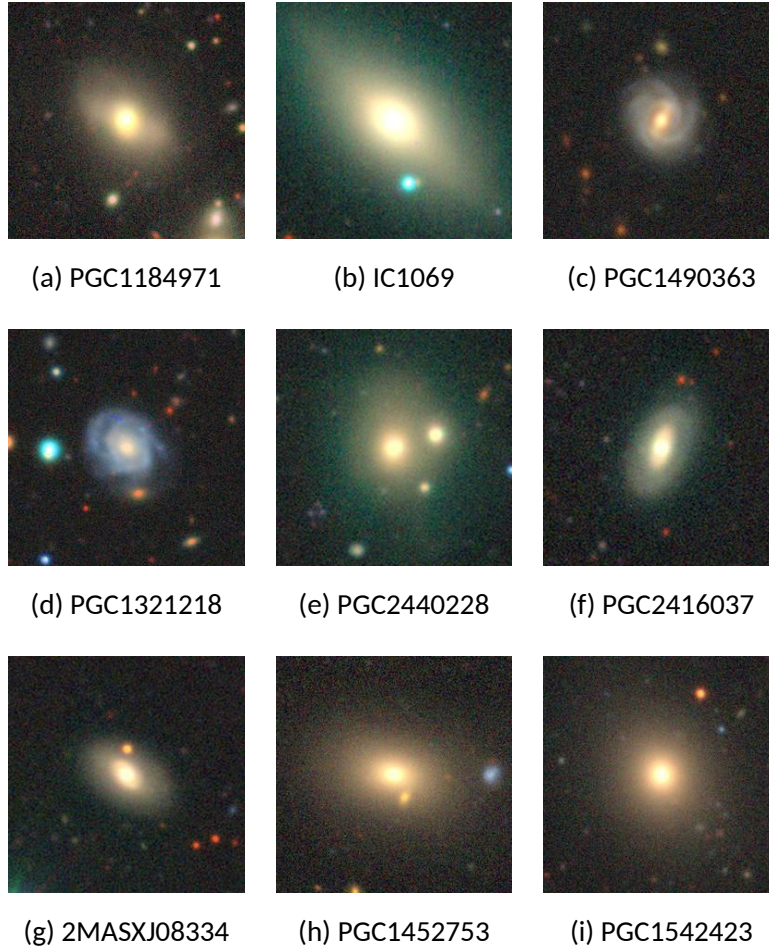


Figura 9: Muestra de imágenes de ejemplo obtenidas mediante la API de *Legacy Survey*.

Cada imagen fue procesada para eliminar cualquier cuerpo luminoso que no se encuentre directamente sobre o detrás de la fuente descrita en el catálogo de SGA. Para esto, se empleó la metodología descrita en la sección anterior. La Figura 10 ilustra el resultado obtenido sobre las imágenes expuestas en la Figura 9 después de aplicar *k-means* con $k = 2$. La Figura 11 ilustra el resultado después de aplicar el método basado en el paquete *ndimage* de *scipy*. Esta función, a diferencia de *k-means*, trabaja de manera espacial sobre la imagen, por lo cual no solamente depende del color del objeto: los objetos que no se encuentren conectados espacialmente obtendrán su propia etiqueta, lo que nos permitirá diferenciar la galaxia de otros elementos en la imagen.

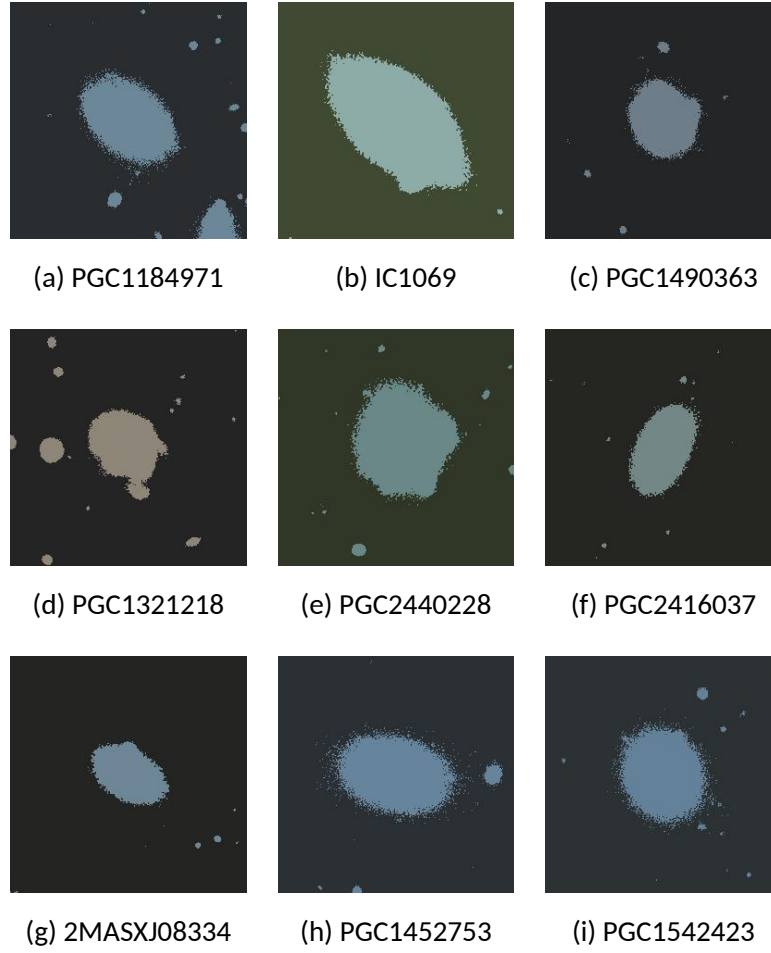


Figura 10: Resultado después de aplicar el método basado en k -means con $k = 2$ para eliminar cualquier cuerpo luminoso que no se encuentre directamente sobre o detrás de la fuente descrita en el catálogo SGA.

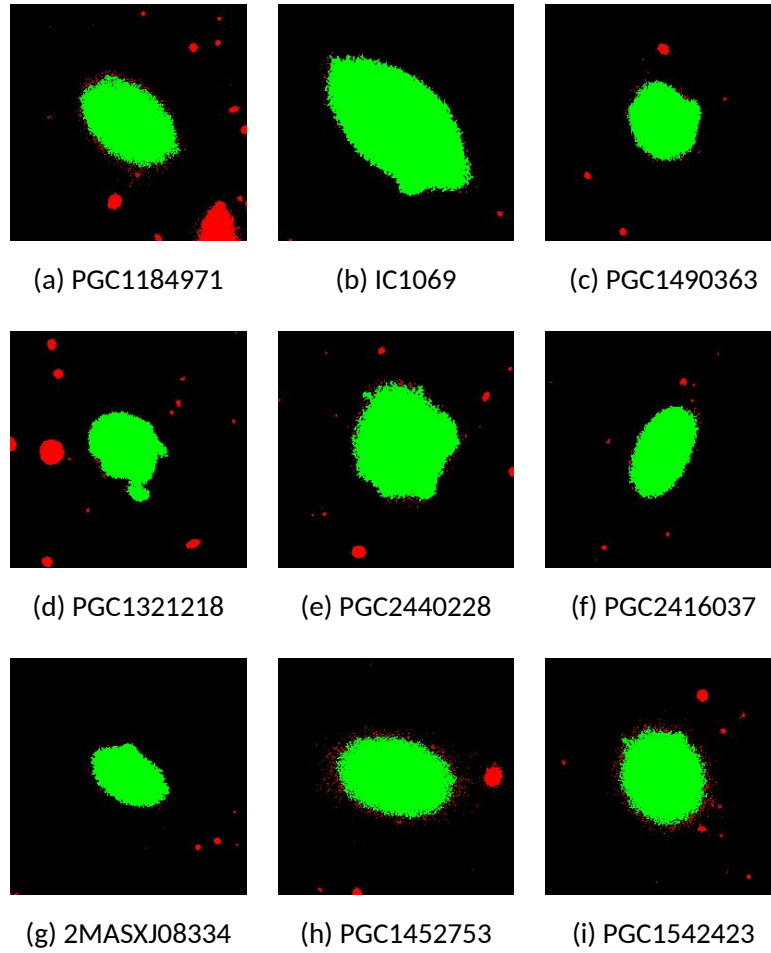


Figura 11: Resultado de aplicar *nlabel* al resultado de k-means: los píxeles de color verde son los que se desea detectar como pertenecientes a la galaxia de interés, mientras se deben eliminar los negros y rojos, correspondientes a otros objetos.

Sabiendo que la imagen del centro es la etiqueta de la fuente de interés, se pueden transformar todos los píxeles que no posean dicha etiqueta a negro o transparente para observar solamente la fuente. La Figura 12 ilustra el resultado obtenido sobre las imágenes de la Figura 9.

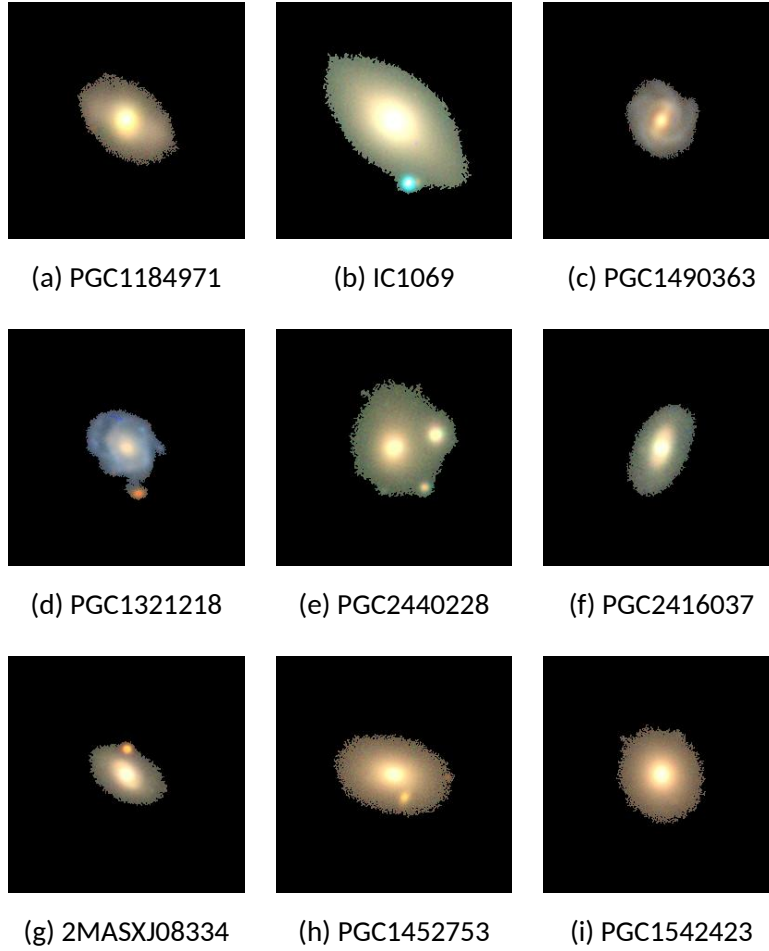


Figura 12: Imágenes originales con objetos extra (en su mayoría) y fondo eliminado.

Cabe notar que todo el proceso descrito trajo una serie de problemas sobre ciertas imágenes que eran muy brillantes, por lo cual se etiquetaba todo como un solo objeto. En otros casos, la galaxia central se salía de los bordes de la imagen. Asegurándose que las esquinas de la imagen no tuvieran la misma etiqueta que la fuente central, se eliminaban estos casos. Después de filtrar estos casos particulares, se obtuvo un conjunto de 135,632 de imágenes de galaxias correctamente identificadas.

Los datos restantes a utilizar del catálogo SGA, además de las posiciones en el cielo, son:

- PA_LEDA: Ángulo de la galaxia en grados.

Galaxy	PA_LED	D25_LED	BA_LED	Z_LED	MAG_LED
PGC1184971	42.52	0.55	0.63	0.08	16.48
IC1069	49.81	1.47	0.52	0.04	14.88
PGC1490363	5.14	0.40	0.85	0.09	16.84
PGC1321218	35.09	0.43	0.67	0.07	16.75
PGC2440228	0.00	0.65	0.68	0.08	16.37
PGC2416037	163.37	0.57	0.50	0.05	16.43
2MASXJ08334	44.19	0.53	0.64	0.05	16.74
PGC1452753	77.42	0.70	0.67	0.09	16.73
PGC1542423	5.81	0.59	0.81	0.08	16.61

Tabla 2: Ejemplos de valores presentes por el catálogo SGA.

- D25_LED: Diámetro del eje mayor en arcominutos.
- BA_LED: Cociente entre el eje menor y el eje mayor de la galaxia.
- Z_LED: *Redshift* de la galaxia.
- MAG_LED: Magnitud de la galaxia en Vega mag.

La Tabla 2 expone algunos ejemplos de valores presentes por el catálogo SGA.

Para procesar los datos, lo primero que se realiza es calcular los ejes de cada galaxia en arco-segundos. Para esto, se divide D25_LED en 2 para obtener el radio mayor, luego se multiplica por 60 para obtener su valor en la medida de arcosegundos (arcsec) obteniendo el radio mayor a . Para obtener el radio menor, se multiplica a por BA_LED lo cual se considera como b .

También es necesario modificar la medida de MAG_LED para que esta sea trabajada en unidades AB mag. Para lograr esto, se tiene que la diferencia entre la magnitud m_{AB} y m_{Vega} para la banda VIS es igual a -0.12 . Por lo tanto, a la columna MAG_LED se le debe restar 0.12 de forma que se obtiene la magnitud m en AB mag.

Finalmente, el valor PA_LED se encuentra en grados, pero debido a que este valor será utilizado para obtener ciertos valores, es necesario transformar los grados a radianes. Esto

Galaxy	a	b	m	PA	z
PGC1184971	16.49	10.45	16.36	0.74	0.08
IC1069	44.17	22.86	14.76	0.87	0.04
PGC1490363	11.94	10.17	16.72	0.09	0.09
PGC1321218	12.92	8.65	16.63	0.61	0.07
PGC2440228	19.37	13.10	16.25	0.00	0.08
PGC2416037	17.14	8.49	16.30	2.85	0.05
2MASXJ08334	15.82	10.12	16.62	0.77	0.05
PGC1452753	20.95	14.03	16.61	1.35	0.09
PGC1542423	17.67	14.36	16.49	0.10	0.08

Tabla 3: Ejemplos de valores transformados del catálogo SGA.

Fuente	Índice de Sérsic
PGC1184971	1.189370543
IC1069	1.032729288
PGC1490363	1.206455682
PGC1321218	1.415270771
PGC2440228	2.727889306
PGC2416037	1.360689922
2MASXJ08334	1.343032082
PGC1452753	1.351743954
PGC1542423	1.516391507

Tabla 4: Resultados de la estimación del índice de Sérsic.

se realiza multiplicando la columna PA_LED A por $\frac{\pi}{180}$ obteniendo PA en radianes.

La Tabla 3 expone algunos ejemplos de valores transformados que se pueden comparar con los correspondientes valores originales expuestos en la Tabla 2.

4.2.2. Estimación y cálculo de parámetros para las fuentes

Como se discutió en la sección anterior, para obtener el parámetro de índice de Sérsic se utilizó la librería *statmorph*. Se llevaron a cabo estimaciones independientes sobre las 135,632 obtenidas en la etapa anterior. La Tabla 4 ilustra el resultado obtenido para las galaxias que hemos venido usando como ejemplo.

Fuente	a	b	z	K	z'	K'	a'	b'
PGC1184971	16.49	10.45	0.08	1.52	0.81	7.53	3.33	2.11
IC1069	44.17	22.86	0.04	0.75	2.82	7.84	4.20	2.17
PGC1490363	11.94	10.17	0.09	1.69	2.23	8.25	2.45	2.09
PGC1321218	12.92	8.65	0.07	1.26	2.22	8.26	1.98	1.32
PGC2440228	19.37	13.10	0.08	1.51	1.18	8.27	3.53	2.38
PGC2416037	17.14	8.49	0.05	0.97	2.10	8.32	1.99	0.99
2MASXJ08334	15.82	10.12	0.05	1.02	2.44	8.11	1.99	1.27
PGC1452753	20.95	14.03	0.09	1.75	3.14	7.60	4.83	3.23
PGC1542423	17.67	14.36	0.08	1.46	2.94	7.75	3.33	2.70

Tabla 5: Resultado de cambio de los valores originales a y b al asignar un nuevo *redshift* z' para obtener a' y b' .

Cabe notar que para 7,186 imágenes de las 135,632 originales no se le logró estimar el índice debido a que la librería no logró obtener un resultado o presentó una advertencia de que la estimación realizada sobre la imagen no era de muy buena calidad. Para evitar en su mayoría errores en los datos, se filtraron estos casos, dejando un total de 128,446 imágenes.

Para aplicar el cambio de *redshift* se requiere aplicar los cambios de escala y luminosidad planteados en 3.1.3, siendo el primer paso modificar la escala de distancias en la cual se encuentran los ejes de las fuentes. Para obtener un mayor rango de imágenes, el proceso de asignar un nuevo *redshift* y ajustar los parámetros considerando los cambios de escala se realizó tres veces sobre las 128,446 fuentes, obteniendo así 385,338 fuentes con distinto *redshift*. Un ejemplo de los resultados del cambio de ejes se puede observar en la tabla 5 siguiendo las galaxias ejemplificadas.

Usando el procedimiento anterior, se obtuvo un promedio de cambio de escala K/K' de 0.15 con una desviación estándar de 0.07, lo que quiere decir que las fuentes en promedio se consideraban con un promedio de 15 % de su tamaño original.

El siguiente paso fue el cálculo de las distancias de luminosidad d_1 y d_2 correspondientes a los *redshift* z y z' . Esto permite realizar el cambio de las magnitudes de las fuentes mediante la ecuación 20.

El siguiente paso fue realizar el cambio de la distancia de luminosidad a la magnitud presente en las fuentes. Para esto se utiliza la ecuación 20. Un ejemplo de aplicar este procedimiento en las galaxias que se han ejemplificado se observan en la tabla 6.

Galaxy	z	d_1 [Mpc]	m [AB mag]	z'	d_2 [Mpc]	m' [AB mag]
IC1069	0.04	165.48	14.88	2.82	23556.84	25.52
PGC1184971	0.08	366.87	16.48	0.81	5071.15	22.06
PGC1321218	0.07	296.25	16.75	2.22	17609.57	25.50
PGC1452753	0.09	432.30	16.73	3.14	26822.71	25.57
PGC1490363	0.09	415.23	16.84	2.23	17739.45	24.87
PGC1542423	0.08	349.03	16.61	2.94	24788.33	25.75
PGC2416037	0.05	219.93	16.43	2.10	16473.21	25.68
PGC2440228	0.08	362.03	16.37	1.18	8067.50	22.99
2MASXJ08334	0.05	232.60	16.74	1.09	7312.76	24.11

Tabla 6: Resultado del cambio de escala sobre la distancia de luminosidad según un nuevo *redshift*.

En este caso, la magnitud pasó de un promedio de 16.88 a un promedio de 24.40. Notar que la magnitud es un valor que se encuentra en una escala logarítmica y definida de tal forma que a mayor valor, menor será la luminosidad de la fuente. Por este motivo, la luminosidad de las fuentes disminuyó notoriamente al aumentar el redshift. Como filtro extra y debido a que se busca simular imágenes obtenibles por el telescopio Euclid, se filtran todos los valores de magnitud mayor o igual a 26, con lo cual el conjunto final de datos termina siendo de 225,010.

Con estos valores y las ecuaciones definidas en 3.1.1 se pueden obtener las variables restantes para la representación del perfil de Sérsic a utilizar en *deeplenstronomy*.

Con esto se obtuvo los valores:

- Índice de Sérsic n_S obtenido de la estimación basada en la imagen de la galaxia
- R_S obtenido utilizando la ecuación 15 utilizando los valores a' y b' obtenidos después del cambio de *redshift*.

- Las componentes e_1 y e_2 utilizando las ecuaciones 18 y 19 respectivamente. Utilizando el PA en radianes y la elipticidad obtenida con los valores de a' y b' y la ecuación 16.
- El valor de la magnitud m' obtenida después de realizar el cambio de redshift en base a los datos de MAG_LEDA.

Galaxy	R_s	n_s	e_1	e_2	m'	z'
IC1069	3.02	1.03	-0.05	0.31	25.52	2.82
PGC1184971	2.65	1.19	0.02	0.22	22.06	0.81
PGC1321218	1.62	1.42	0.07	0.19	25.50	2.22
PGC1452753	3.95	1.35	-0.18	0.08	25.57	3.14
PGC1490363	2.26	1.21	0.08	0.01	24.87	2.23
PGC1542423	3.00	1.52	0.10	0.02	25.75	2.94
PGC2416037	1.40	1.36	0.28	-0.19	25.68	2.10
PGC2440228	2.90	2.73	0.19	0.00	22.99	1.18
2MASXJ08334	1.58	1.34	0.01	0.22	24.11	1.09

Tabla 7: Resultado del cálculo y estimación de los parámetros del perfil de Sérsic para ser utilizados como fuente bajo el efecto de lente.

Finalmente, la posición en la cual se encuentran estas fuentes dentro de la imagen se añadió aleatoriamente en un rango de $[-2, 2]$, tanto para el eje x como el eje y. Este valor fuerza a que la probabilidad de que ocurra *strong lensing* muy alta. Notar que esta posición se encuentra en arco-segundos y la imagen final tiene el rango aproximado de $[-5, 5]$ para cada eje.

4.2.3. Parámetros de instrumento, lente y ruido

Como se describió en la sección 2.2.1, el archivo de configuración de *deeplenstronomy* requiere una serie de parámetros para definir la imagen en *IMAGE* y definir el instrumento en *SURVEY*. Estos parámetros son definidos basándonos en lo que señala la librería *lenstronomy* que a su vez se basa en el trabajo de preparación para los datos del telescopio Euclid [Euclid Collaboration et al., 2022]. Estos valores corresponden a la tabla 8.

Parámetro	Valor
<i>exposure_time</i>	565.0 [s]
<i>numPix</i>	101 [pixel]
<i>pixel_scale</i>	0.1 [arcosegundo/pixel]
<i>psf_type</i>	GAUSSIAN
<i>read_noise</i>	4.2 [electrones]
<i>ccd_gain</i>	3.1 [electrones/ADU]
<i>BANDS</i>	VIS
<i>seeing</i>	0.16 [μm]
<i>magnitude_zero_point</i>	30.84 [AB Mag]
<i>sky_brightness</i>	22.35 [AB Mag]
<i>num_exposures</i>	4

Tabla 8: Parámetros y valores

Debido a que no se encuentran disponibles modelos específicos de objetos *oscuros* para poder basar la simulación de una lente, los parámetros de la lente oscura se definieron basándose en distribuciones con valores entregados por un astrónomo experto.

En el caso de la lente, se requieren los parámetros e_1 , e_2 , σ_v y el *redshift* z_l . Para el caso de las componentes de la elipticidad, se generó de manera uniforme la elipticidad de un objeto con el rango continuo $[0, 0.7]$, rango en el cual se encuentra la componente esferoidal de la lente, de acuerdo con el experto. Además, se generó de la misma forma el ángulo de este cuerpo en el rango $[0, \pi]$, en radianes. Con estos valores y utilizando las ecuaciones 18 y 19, se obtuvieron los valores para la simulación.

Con respecto a σ_v , este valor lo utiliza *lenstronomy* para obtener el valor de la masa de la lente, la cual define cómo será la distorsión de la luz. Siguiendo las indicaciones del experto, se definió una distribución normal con media 250 y desviación estándar de 50, con el fin de obtener valores que se consideren realistas y dentro de los rangos normalmente presentes en la realidad.

Por último, para el *redshift* z_l , se define aleatoriamente de manera uniforme en el rango $[0.5, z' - 0.4]$, donde z' corresponde al *redshift* de la fuente que será sometida al efecto de lente en cada caso. Esta definición se obtuvo del desafío en donde el mínimo de diferencia

entre la lente y la fuente que utilizaron fue de 0.4.

Con el objetivo de que el modelo no aprenda de manera trivial a contar la cantidad de fuentes brillantes en la imagen, se añaden imágenes extra y ruido de Poisson utilizando el procedimiento detallado en la Sección 3.1.4.

4.2.4. Resultado de la simulación de efecto de lensing

Imágenes con *lensing*

Con los datos obtenidos y procedimientos definidos previamente, se utilizó la librería *deeplenstronomy* para lograr obtener imágenes de efecto de *lensing*. Estas imágenes contienen entre una a tres fuentes, donde solo la fuente con mayor *redshift* (o más lejana) se ve afectada por un lente que no es visible en la imagen. La Figura 13 ilustra algunos de los resultados obtenidos.

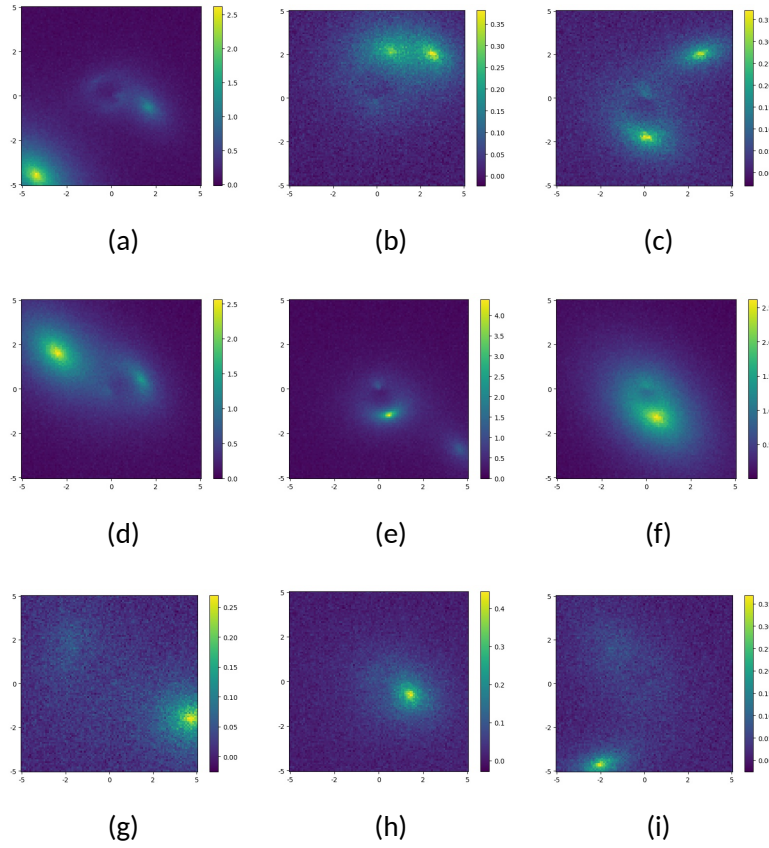


Figura 13: Efectos de *lensing* en escala Viridis (ver texto principal para una mayor discusión).

En los resultados se pueden observar una serie de casos con claros efectos de *lensing* como en la imagen (b) de la Figura 13. También se observan casos en donde la lente se ve opacada por las galaxias extra que presenta la imagen, como en los casos (a), (g), y (i). Cabe destacar que la visualización de estas imágenes para seres humanos es un problema a resolver por sí mismo. En estos ejemplos, se presentó una imagen en la cual los colores se eligen en función del valor máximo y mínimo, lo cual puede generar casos donde ciertos cuerpos no son observables para una persona, pero sí se encuentran visibles si se analizan los valores numéricos.

Durante la inspección de los resultados por parte del experto, se destaca una gran variedad de configuraciones visibles, como por ejemplo un anillo de Einstein, el cual aparecía en la

muestra con una frecuencia aproximada de 3 cada 100 casos, que si bien es baja frecuencia, se encuentra un orden de magnitud mayor debido a las posiciones que se les asignó a las fuentes. La Figura 14 presenta un ejemplo de este peculiar y relevante caso de lensing. Sin embargo, también se detectaron problemas en las imágenes, particularmente un problema con la cantidad de luz que tienen las galaxias. Más específicamente, se observa que estas tienen una zona en la cual la luminosidad es mucho más baja que en el centro, zona que no es usualmente visible en imágenes reales. La Figura 15 ilustra ejemplos de esta anomalía.

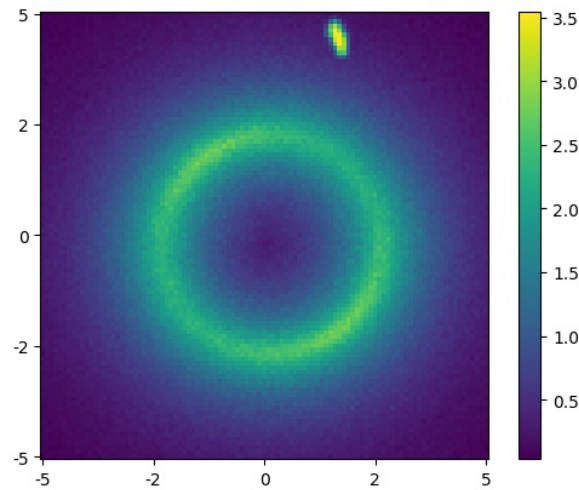


Figura 14: Ejemplo obtenido de anillo de Einstein.

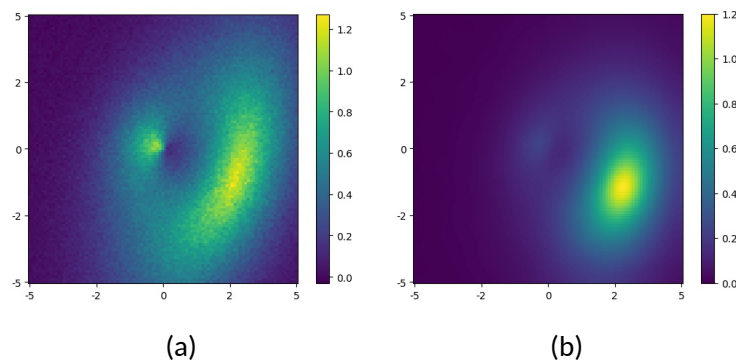


Figura 15: Efectos de *lensing* con mucha luz.

En 15, se puede observar como el brillo de las galaxias es muy amplio, al punto que las imáge-

nes múltiples de la misma galaxia generadas por efecto del *lensing* se encuentran conectadas por un "puente luminoso" menos notorio, pero que no debería poder ser observable. Este fenómeno ocurre en una proporción aproximada de 9 cada 100 casos.

Otro problema destacado en las imágenes corresponde a casos en los cuales la segunda imagen de la galaxia crea un efecto de "puente luminoso" con la posición de la lente (que se encuentra en el centro de las fotos).

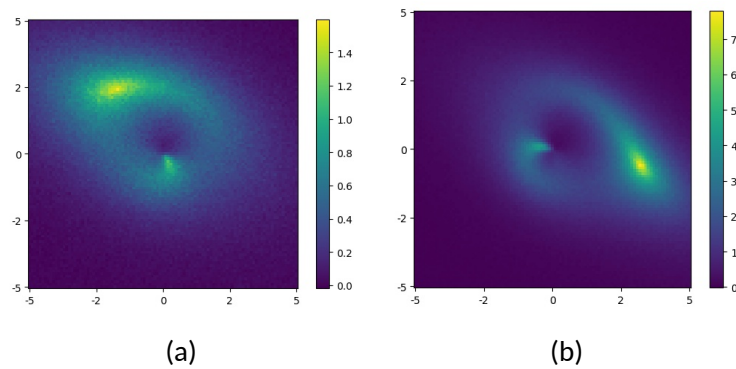


Figura 16: Efectos de lensing con "puente".

Que la posición más brillante sea este "puente" es una anomalía. Estos casos ocurrían en una proporción de 6 de cada 100, lo que hace que su frecuencia observada sea un poco más alta de lo habitualmente predicho por la teoría. Este "puente brillante" es una réplica de la imagen de la galaxia, con un tamaño menor en comparación con la imagen opuesta y una distorsión de tipo radial. Dado que estas imágenes requieren una configuración especial, no se observan con tanta frecuencia, por lo que estos sistemas están ligeramente sobre-representados.

Los dos problemas descritos anteriormente no son excluyentes, es decir, puede ocurrir que una galaxia posea tanto un puente como una difusión lumínica grande. Si bien estos casos no deben ser despreciados, poseen una baja proporción con respecto a la cantidad total de las imágenes.

Imágenes sin *lensing*

Para generar los casos sin el efecto de lente, se realizó el mismo procedimiento que el caso anterior, pero se asignó un valor de $\sigma_v = 1$ al perfil de masa de la lente, por lo cual se tiene un cuerpo con una cantidad de masa despreciable. Esto, junto a que la magnitud del objeto toma un valor despreciable (un valor muy alto de magnitud), hace que no se observe ni la lente, ni el efecto de esta en la foto obtenida.

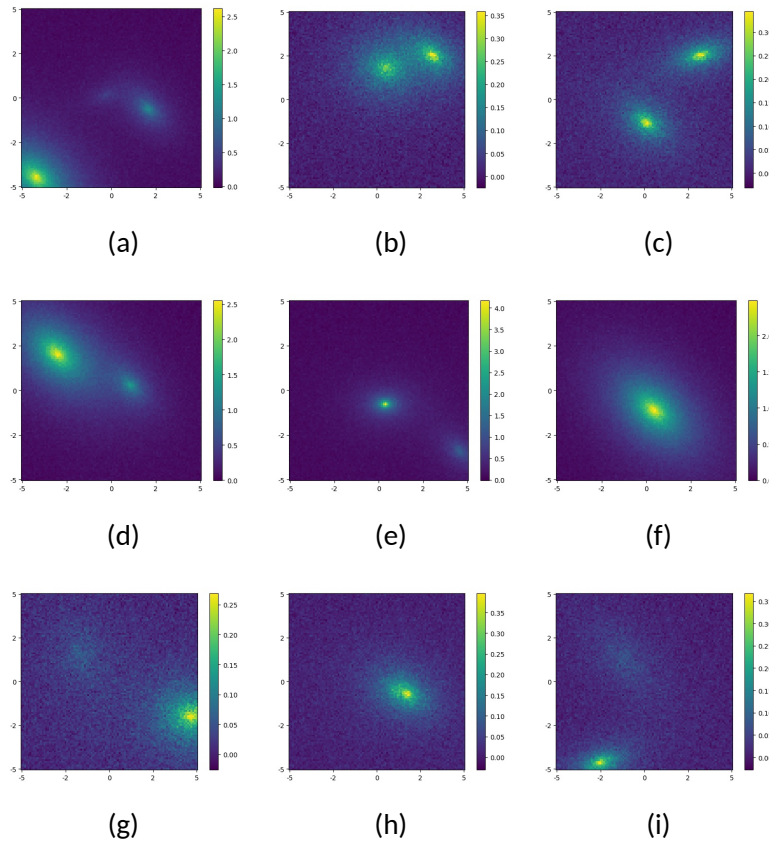


Figura 17: Imágenes de ejemplo sin *lensing* en escala viridis.

4.3. Modelo de *deep learning*

Para la implementación de este modelo se implementó el modelo 8, pero se modificó la capa de salida para que esta fuera capaz de retornar dos o tres valores dependiendo del

experimento que se estuviera utilizando. Consistentemente con esto, se modificó la función de pérdida a *categorical crossentropy* tanto para casos con 2 salidas como 3 salidas.

Con respecto a la cantidad de datos a utilizar, este se definió basándose en la clase que presentara la menor cantidad de fotos, de modo de obtener una muestra balanceada. Como la generación de efecto de lente presentada en este trabajo permite generar una cantidad virtualmente ilimitada de datos, el límite viene de los datos liberados para en el desafío de la Universidad de Bologna, descrito en la Sección 2.3. Este *dataset* contiene 20,000 imágenes, donde 13,968 corresponden a casos de efecto de lente brillante y 6,032 corresponden a casos de no lente. Por lo tanto, es esta última clase la cual pone el límite a la cantidad de datos totales a utilizar. Como se detalla más adelante, estas limitaciones y consideraciones producen un total de imágenes igual a 36,192.

De estos 36,192 datos, 80 % fue utilizado para el entrenamiento y el 20 % restante fue utilizado para las pruebas. Para el entrenamiento de todos los modelos se utilizaron 125 *epochs* con un *batch* de 32 por limitaciones de procesamiento.

4.3.1. Experimento 1

Para la primera estrategia de entrenamiento (ver Sección 3.2.1), solamente se utilizaron dos de las cuatro clases de datos disponibles: 18,096 casos de *lensing oscuro* y 18,096 imágenes sin efecto de lente.



Figura 18: Función de perdida a lo largo de las *epochs* de entrenamiento para el Experimento 1.

Para este caso, se utilizaron 28,951 imágenes de entrenamiento. Se observa que en la figura 18, el error de entrenamiento de la red fue decayendo a medida que transcurría la optimización, hasta aproximadamente la época 100, en la cual ya no se observa una disminución notoria y simplemente varía de forma aleatoria al continuar la optimización. Eso puede haber generado un cierto grado de *overfitting*, que en caso de suceder se debería observar al medir el desempeño en el conjunto de pruebas.

En la fase de evaluación o *pruebas*, se utilizaron las 7,238 imágenes restantes. Cabe enfatizar que la red no tuvo acceso a ellas en ningún momento del entrenamiento. Se puede observar de la figura 19 que la red no tuvo mayores problemas para diferenciar las imágenes de lente *oscuro* con los casos sin lentes.

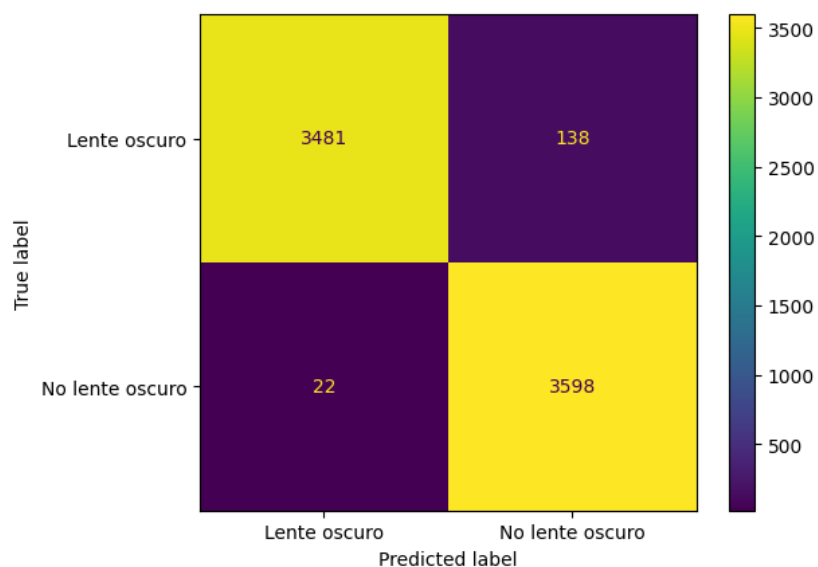


Figura 19: Matriz de confusión para el Experimento 1.

Donde más se equivoca el modelo es en los falsos negativos, es decir, donde predice que no hay lente cuando si la hay. En la figura 20 se observa que este error se debe a que el efecto de *lensing* en las imágenes corresponde a uno débil o la imagen de la fuente a la cual se le está aplicando *lensing* se encuentra opacada.

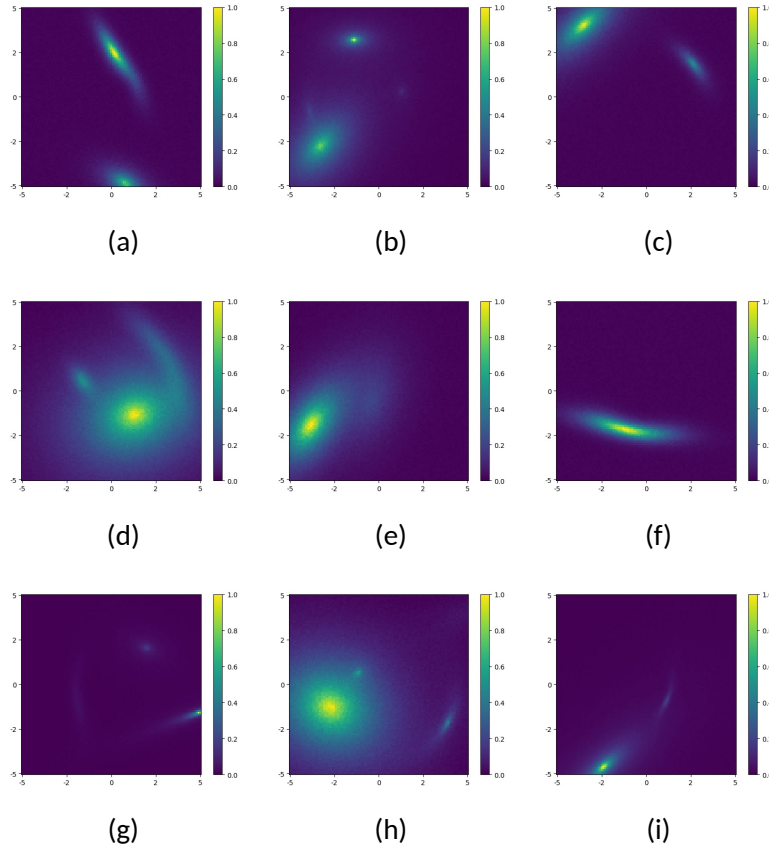


Figura 20: Casos de ejemplo en los cuales el modelo confundió efectos de lente *oscuro* como casos de sin lente en el Experimento 1.

Por el otro lado, los casos de falsos positivos, se observa en la figura 21 que el supuesto efecto de lensing que observa el modelo se debe a la posición en la cual se encuentran las galaxias "simulando" el efecto de *lensing*, pero donde no se observa una deformación.

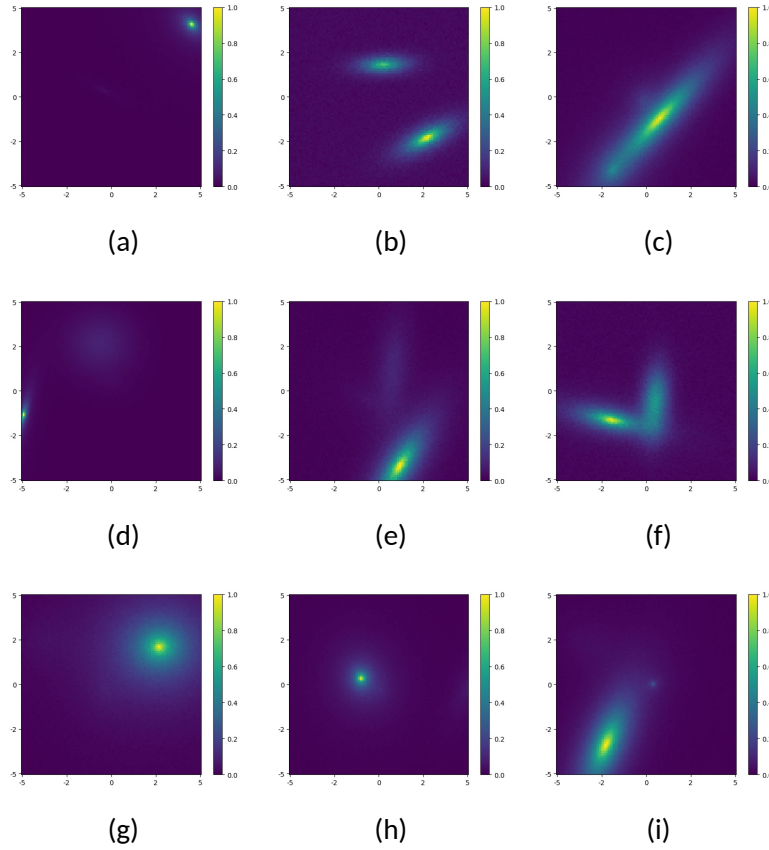


Figura 21: Casos de ejemplo en los cuales el modelo predijo efecto de lente cuando este no está presenta en la imagen en el Experimento 1.

Por otro lado, el modelo obtuvo una exactitud (*accuracy*) de 0.978 y, como se resume en la tabla 9, excelentes valores en todas las métricas analizadas. Sin desmedro de lo anterior, se puede observar que el *recall* es el menor valor de entre los datos que corresponden a la clase lente oscura. Esto lleva a concluir que la mayor limitación relativa del modelo consiste en detectar menor cantidad de lentes *oscuras* en comparación con los que realmente existen en el conjunto de pruebas (falsos negativos). Esto es consistente con las métricas que se obtienen al evaluar la clase no lente como clase positiva.

Clase	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Lente oscuro	0.99	0.96	0.98	3619
No lente	0.96	0.99	0.98	3620

Tabla 9: Reporte de clasificación para el Experimento 1.

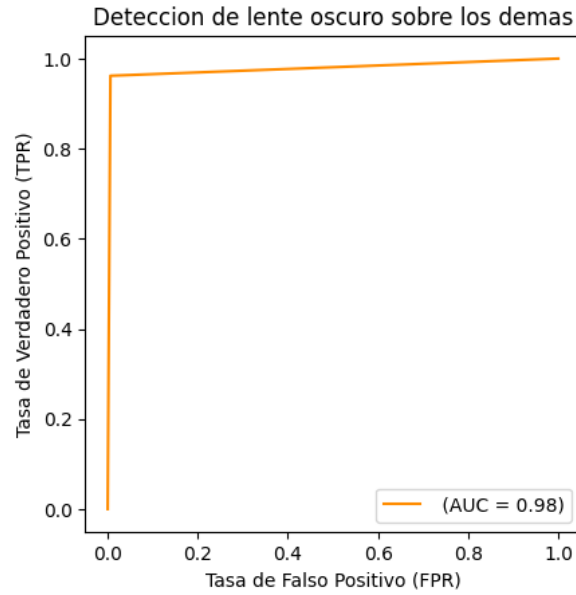


Figura 22: Curva ROC obtenida al comparar la predicción de lentes oscuros vs no lente (Experimento 1).

Finalmente, para este conjunto de datos se obtiene la curva ROC y su área (AROC) de 0.98, valor que se encuentra cercano a los resultados obtenidos por el modelo *Lensfinder* en el desafío de la Universidad de Bologna, descrito en la Sección 2.3.

4.3.2. Experimento 2

Para la segunda estrategia de entrenamiento, se utilizaron tanto imágenes de lente oscuro como imágenes del GLFC (*Gravitational Lens Finding Challenge*) [Metcalf et al., 2019]. Las consideraciones detalladas en la Sección 3.2.1, llevan a las siguientes proporciones de datos:

- Lente oscuro generados en este trabajo: 18,096 casos.
- Lente brillantes del GLFC [Metcalf *et al.*, 2019]: 6,032 casos.
- Sin lente generados en este trabajo: 6,032 casos.
- Sin lente del GLFC [Metcalf *et al.*, 2019]: 6,032 casos.

Como los casos de lentes brillantes y sin lente tanto del GLFC como creadas en este trabajo son consideradas en la misma categoría, se balancearon las cantidades de acuerdo a la subclase que ponía la mayor limitación.

Al igual que en el experimento anterior, se utilizó un 80 % de datos para el entrenamiento balanceado, es decir, 28,951 imágenes. Sin embargo, a diferencia de las pruebas anteriores, la cantidad de imágenes sin *lensing* creadas por *deeplenstronomy* disminuyó para darle espacio a las imágenes del GLFC.

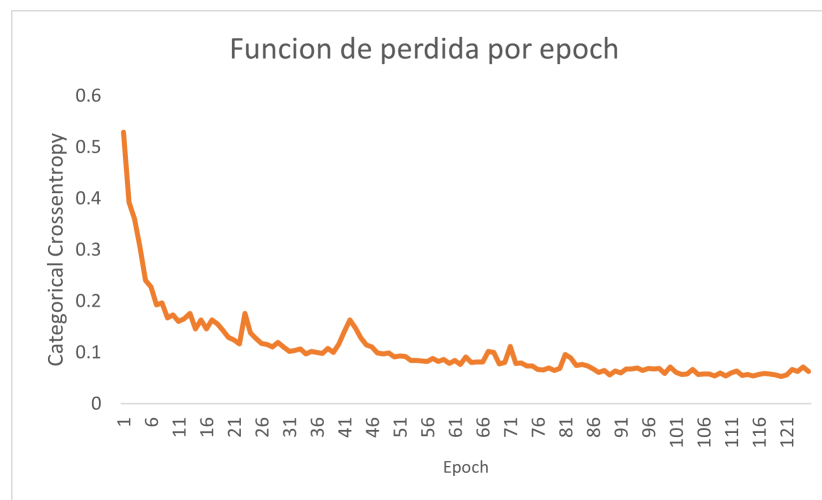


Figura 23: Función de pérdida a lo largo de las *epochs* de entrenamiento para el Experimento 2.

Los resultados del entrenamiento fueron parecidos a los observados en el primer experimento, con una velocidad de decaimiento relativamente más lenta. En este caso, fue necesario

optimizar más tiempo el modelo (mayor cantidad de *epochs*) para llegar a la asíntota de entrenamiento.

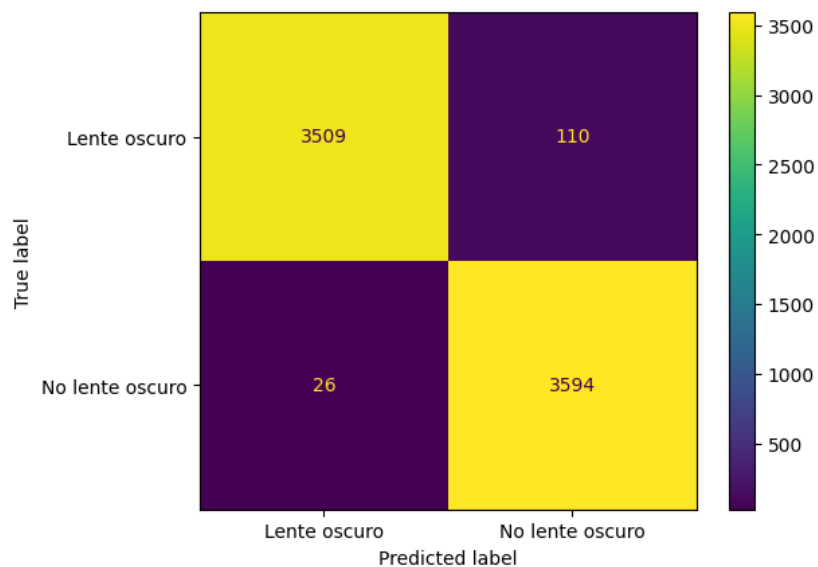


Figura 24: Matriz de confusión para el Experimento 2.

Nuevamente, se observa en la figura 25 que la presencia de casos *lensing oscuro* que no son detectados por el modelo es debida a que el efecto de este corresponde principalmente a un caso de distorsión.

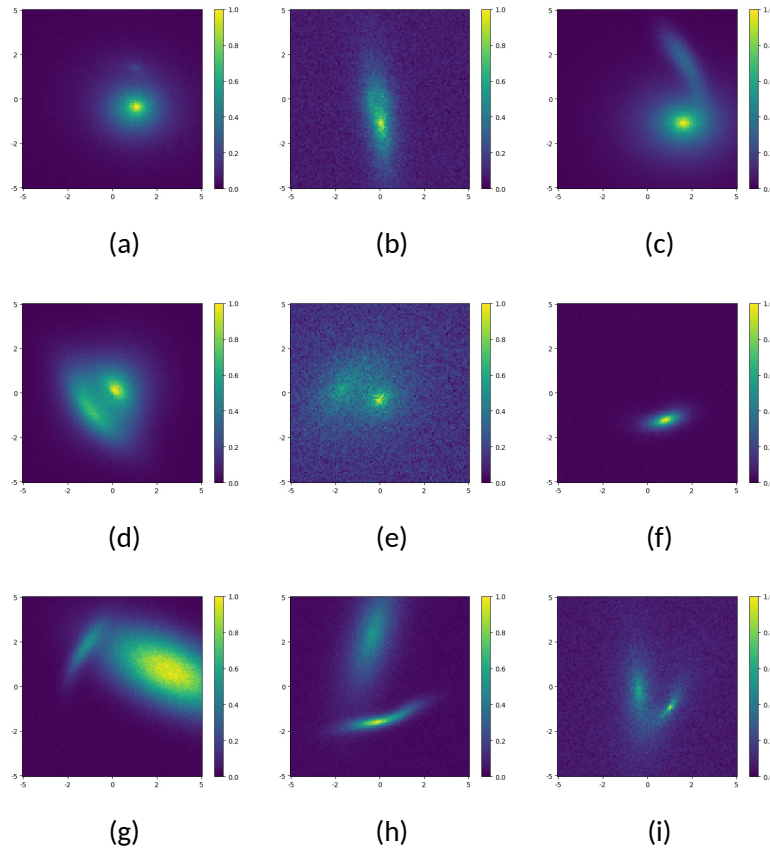


Figura 25: Casos de ejemplo en los cuales el modelo no pudo detectar el efecto de *lensing* oscuro en el Experimento 2.

En la figura 26 que la posición de las fuentes hace parecer efecto de que existe efecto de *lensing* debido a sus posiciones. Además, se destaca en el caso 26h que si ocurre el efecto de lente, sin embargo, este ocurre debido a una galaxia brillante.

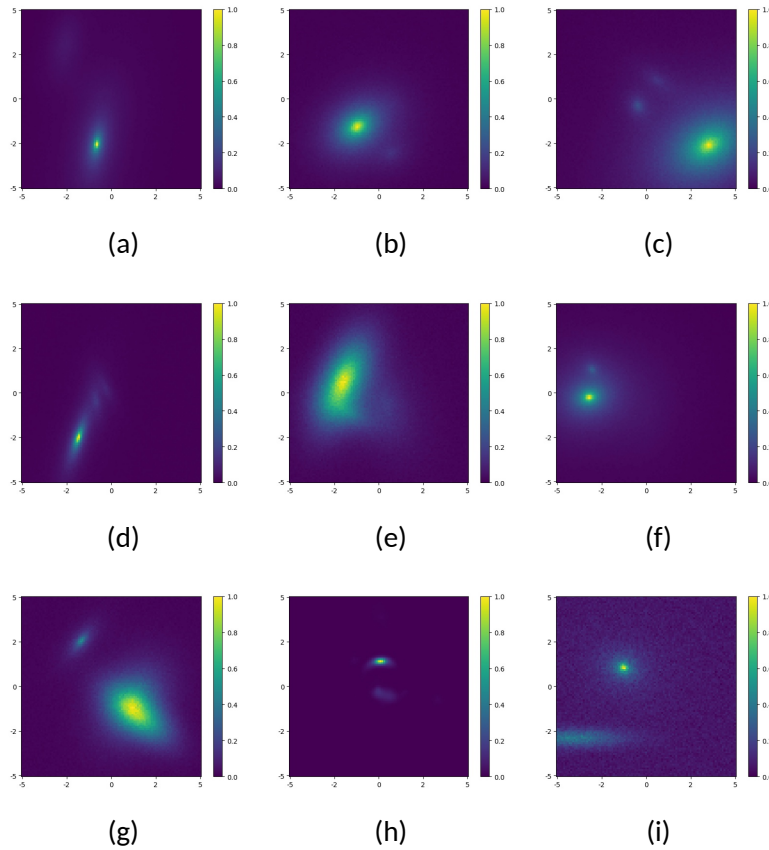


Figura 26: Casos de ejemplo en los cuales el modelo observa un efecto de *lensing oscuro* donde no existe en el Experimento 2.

Gracias a 24, se puede ver que al añadir los casos tanto de lente brillante como no lente provenientes del GLFC, no se observa un cambio notorio en los resultados con respecto a utilizar solamente los ejemplos negativos creados en este trabajo. De hecho, los resultados obtenidos son algo mejores, con una exactitud global (*accuracy*) de 0.981.

Clase	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Lente oscuro	0.99	0.97	0.98	3619
No lente Ocuo	0.97	0.99	0.98	3620

Tabla 10: Reporte de clasificación para el Experimento 2.

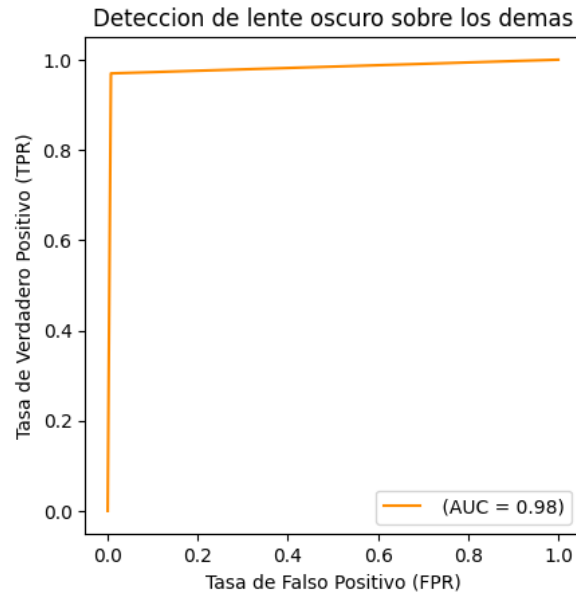


Figura 27: Curva ROC obtenida al comparar la predicción de lentes oscuros vs no lente y lente brillante (Experimento 2).

La curva ROC obtenida (y, por lo tanto, el AUROC) es idéntica a la obtenida en el experimento 1.

4.3.3. Conjunto de datos N°3

En este último experimento, se utilizan tanto imágenes de lente *oscuro* como imágenes de lente convencional provenientes del GLFC [Metcalf *et al.*, 2019]. Sin embargo, para estas pruebas las imágenes de lentes brillantes configuran una categoría diferente. Las consideraciones detalladas en la Sección 3.2.1, llevan a las siguientes proporciones de datos:

- Lente *oscuro* generados en este trabajo: 12,064 casos.
- Lente brillantes del GLFC [Metcalf *et al.*, 2019]: 12,064.
- Sin lente generados en este trabajo: 6,032 casos.
- Sin lente del GLFC [Metcalf *et al.*, 2019]: 6,032 casos.

Los datos fueron balanceados de tal forma que las tres clases que debe reconocer el modelo (lente oscuro, lente brillante y no lente) tengan la misma representatividad.



Figura 28: Función de pérdida a lo largo de las *epochs* de entrenamiento para el Experimento 3.

En este experimento, el modelo tuvo una mayor dificultad para aprender. Como se observa en 28, la función de pérdida decae mucho más lentamente y termina en un valor más elevado, cercano al doble del valor de mínimo error encontrado en ambos experimentos previos.

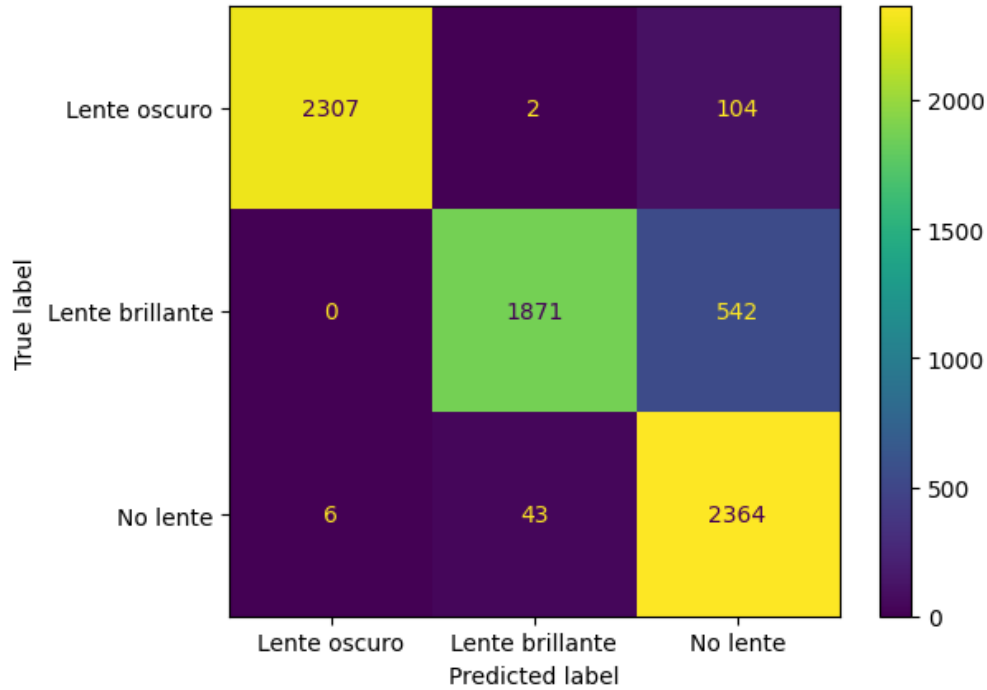


Figura 29: Matriz de confusión para el Experimento 3.

En la figura 30 se pueden observar los casos para cada categoría de predicción. En estas se identifica en 30b como de uno de los dos casos en la cual confunde una lente *oscura* con una lente brillante, es debido a la configuración en la cual se encuentra la galaxia distractora dentro de la imagen. En el caso donde se comete la mayor cantidad de errores es donde muchas imágenes de lente brillante son clasificadas como casos sin *lensing*, como en 30e donde no puede observar el efecto de *lensing*, como una imagen distorsionada o múltiples imágenes de alguna fuente. Existe la probabilidad que las imágenes que generadas por el *lensing*, ya sea distorsionada y/o múltiples imágenes, sea tenue en comparación a la luz de la lente.

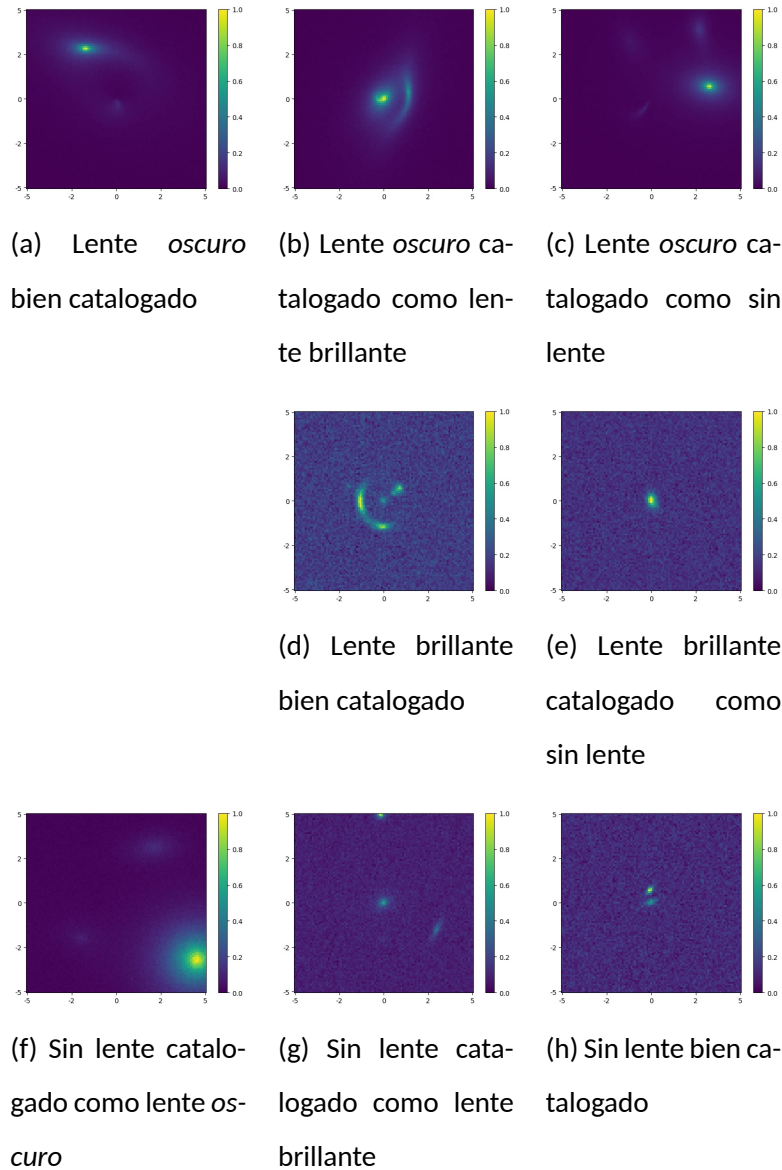


Figura 30: Ejemplo de cada categoría observable en la figura 29. Notar que no existe el caso que se predice lente *oscuro* cuando el caso es de lente brillante en el Experimento 3.

Al evaluar el modelo en el conjunto de pruebas, se observa que el desempeño en la detección de lentes *oscuras* no tuvo una mayor diferencia en este experimento. Sin embargo, se puede observar al momento de considerar los casos de lentes brillantes, muchos fueron clasificados como no lente. Por otro lado, la clasificación de los casos que no corresponden a lensing se

mantuvo en un rango muy satisfactorio.

Clase	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
Lente oscuro	1	0.96	0.98	2413
Lente brillante	0.98	0.78	0.86	2413
No lente	0.79	0.98	0.87	2413

Tabla 11: Reporte de clasificación para el Experimento 3.

La exactitud (*accuracy*) global obtenida fue 0.90. De manera consistente con la discusión anterior y como se puede observar en la Tabla 11, el *Recall* para la clase de lente brillante es la métrica menos satisfactoria junto con la *Precision* para la clase de no lente (si bien se clasificaron correctamente la gran mayoría de casos que no son lentes, hubo una cantidad no menor de casos que fueron clasificados como no lente cuando si poseían el efecto de lente).

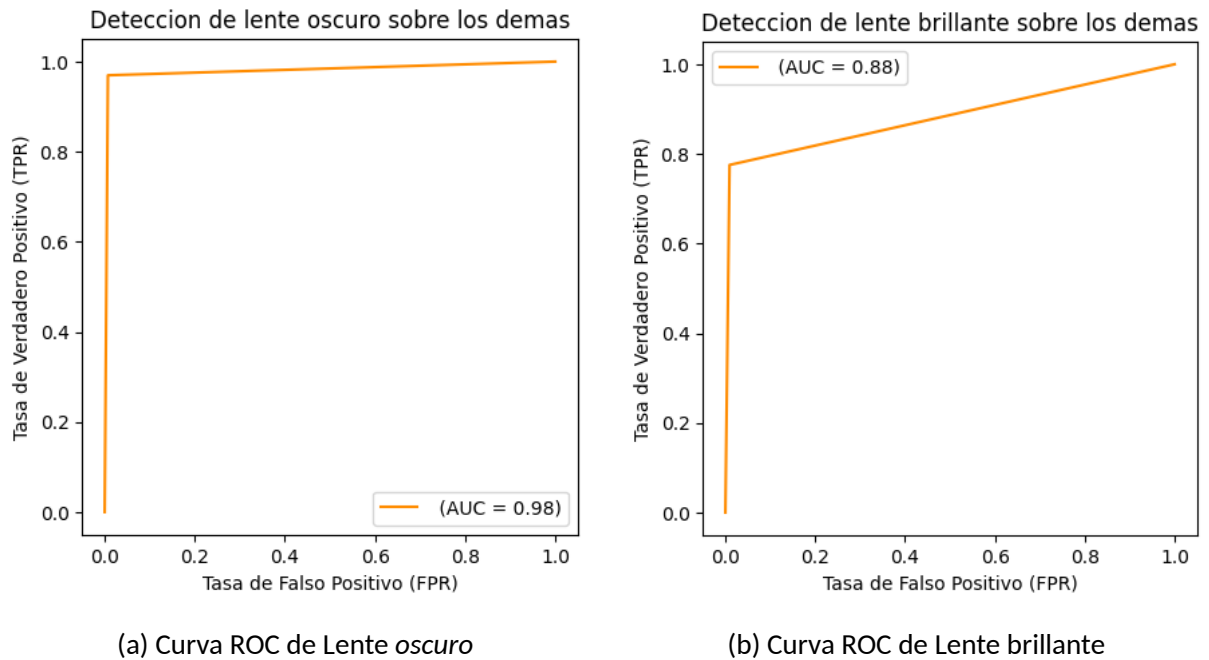


Figura 31: Curva ROC para las dos clases de lentes consideradas en el Experimento 3.

Observando las gráficas de ROC, se observa que no hubo efecto en la predicción de lentes

oscuros, manteniendo un AUC de 0.98, mientras que en la curva de lente brillante se observa un peor resultado, con un AUC de 0.88.

CAPÍTULO 5

CONCLUSIONES

En este capítulo se presentan las conclusiones con respecto al trabajo y pruebas realizadas, además del posible trabajo futuro y mejoras posibles a realizar.

5.1. Generación de imágenes

5.1.1. Estimación del índice de Sérsic

La metodología propuesta para la simulación de casos de lensing, requiere generar un perfil de luminosidad Sérsic consistente con galaxias reales. Sin embargo, el índice de Sérsic no se encontraba disponible en el catálogo utilizado (*Legacy Survey*). Para esto, se propuso una metodología que permite estimar este dato faltante y que requiere una imagen que contenga solo la galaxia aislada de otros distractores luminosos.

El método que se diseñó para lograr obtener las imágenes de las galaxias aisladas consiguió 135,632 imágenes de 160,526 casos, lo cual es un buen porcentaje. Sin embargo, es importante tener en consideración que este método no elimina los casos que la fuente poseía otra galaxia encima (en términos de imagen), lo cual tendrá un efecto al momento de utilizarlas en la estimación.

Sobre las 135,632 imágenes pre-procesadas, la librería *statmorph* logró ajustar un perfil Sérsic exitosamente en todos, excepto en 7,186 casos. Estos casos fallidos podrían corresponder a limitaciones conocidas de la librería o haberse visto afectados por limitaciones del preprocesamiento propuesto. Finalmente, es importante señalar que los parámetros que presenta *Legacy Survey* también son estimados y que podría existir una mala compatibilidad entre el método utilizado para estimar estos parámetros y el método propuesto para estimar el

índice de Sérsic.

5.1.2. Cambio de *redshift*, ejes y magnitud

La metodología propuesta para la modificación artificial del *redshift* de galaxias reales, logró generar un conjunto de datos más variado. Sin embargo, se observó que la transformación de las imágenes afectó significativamente el tamaño de las galaxias, ya que se redujo en un promedio del 85 % su tamaño original. Si bien este cambio se hizo considerando los parámetros cosmológicos, no se debe desestimar el posible sesgo que este cambio puede haber generado. Un símil ocurre con el cambio de magnitud, lo cual puede haber generado algún otro sesgo con respecto a la luminosidad de las imágenes.

5.1.3. Imágenes simuladas con *deeplenstronomy*

El resultado de las imágenes generadas con la metodología descrita fue mayoritariamente positiva. Se logró obtener una serie de casos en las cuales se observaban correctamente distintos fenómenos de *lensing* que también pueden ser observados por la realidad. Después de perfeccionar el método hasta su versión final, unas pocas centenas de imágenes creadas con el método definitivo fueron revisadas por un astrónomo experto, quien las analizó y proporcionó las consideraciones descritas en el 3.2.2.

Además de las muchas imágenes realistas que fueron aceptadas, también se destaca que se obtuvieron una serie de imágenes clasificadas como “poco realistas” por el experto, en las cuales se observaban distribuciones de luz poco comunes. También se observaron diversos casos en los cuales la cantidad de luz que las imágenes poseían estaba distribuida de manera muy dispersa. Este último problema puede haberse generado debido a que el método de visualización no era adecuado. Otra explicación posible es que los parámetros (en particular el radio de Sérsic) obtenidos del catálogo SGA y la estimación del índice de Sérsic no fueran compatibles, generando así una imagen en la cual la luz de la galaxia no se encontraba tan

concertada en el centro, extendiendo y difuminando los bordes de la fuente.

5.2. Modelo de *deep learning*

5.2.1. Experimento 1

Los resultados de la primera estrategia de entrenamiento sugieren que el modelo es capaz de diferenciar correctamente los efectos de *lensing oscuro* sin una gran dificultad, obteniendo un AUC de 0.98. Este resultado se puede deber a que el modelo efectivamente puede detectar los patrones subyacentes a un caso de *lensing oscuro* o simplemente a que las imágenes de ambas clases eran muy fáciles de diferenciar por artefactos de la simulación. Lamentablemente, ya que no se tienen datos reales de *lensing oscuro*, no es posible distinguir entre estas dos situaciones de modo definitivo.

5.2.2. Experimento 2

Del segundo conjunto de datos se pudo observar que al añadir los datos del *Gravitational Lens Finding Challenge* [Metcalf et al., 2019] no tuvo efecto en la habilidad del modelo en diferenciar un caso de *lensing oscuro* con cualquier otro. Este resultado se puede deber a que el modelo efectivamente puede distinguir los patrones subyacentes a un caso de *lensing oscuro* o de aquellos correspondientes a *lensing tradicional*. También es posible que las imágenes de ambas clases sean muy fáciles de diferenciar por artefactos de la simulación presentada en este trabajo o en [Metcalf et al., 2019]. Nuevamente, ya que no se tienen datos reales de *lensing oscuro*, no es posible distinguir entre estas dos situaciones de modo definitivo.

5.2.3. Experimento 3

En el último experimento, se pudo observar una mayor complejidad de la red al momento de definir la categoría de lente convencional (brillante). Es de esperar que en este caso su rendimiento empeorara, ya que esta vez el modelo estaba encargado de cumplir tres tareas en vez de dos, ya que debía ser capaz de diferenciar efecto de *lensing* en un sistema *oscuro* y brillante.

Nuevamente, ocurre que no tiene problemas con las imágenes de lente *oscuro*, por lo cual es aceptable concluir que el modelo se demuestra efectivo en la tarea para la que fue construido.

5.3. Trabajo futuro

Como trabajo futuro quedan las tareas:

- Utilizar un catálogo de fuentes en el cual se hayan estimado directamente todas las características morfológicas del perfil de Sérsic, de forma que exista una concordancia entre las distintas variables.
- Generar un conjunto de casos lentes convencionales (brillantes) usando el mismo pipeline de simulación utilizado para generar los casos de lentes oscuros con el objetivo de minimizar las diferencias sistemáticas (e.g. características del ruido) entre ambos conjuntos de imágenes y poder establecer con mayor confianza que el modelo está aprendiendo las características específicas de ambas clases de fenómenos.
- Utilizar candidatos reales de *lensing* oscuro como base de validación de la metodología de generación de imágenes y de los modelos de *deep learning* encargados de reconocer este fenómeno.
- Implementar un sistema detector que utilice un clasificador como el definido en este

trabajo, con tal que sea capaz de encontrar efecto de *lensing* a lo largo de la imagen y no requiera que el efecto ocurra centrado.

ANEXOS

5.1. Resumen de unidades

$$[a] = [a'] = [b] = [b'] = [R] = [R_S] = \text{arcsec} \quad (22)$$

$$[K] = [K'] = \text{length/arcsec}(e.g., \text{length} = \text{kpc}) \quad (23)$$

$$[\phi] = \text{rad} \quad (24)$$

$$[m] = [m_{ZP}] = \text{ABmag} \quad (25)$$

$$[I] = [I_0] = \text{count/seconds} \quad (26)$$

5.2. Ejemplo archivo de configuración de *deeplenstronomy*

```
1  DATASET:
2      NAME: ExampleDataset
3      PARAMETERS:
4          SIZE: 3
5          OUTDIR: ExampleDataset
6      SEED: 10
7  COSMOLOGY:
```

```
8     PARAMETERS:
9         H0: 70.0
10        Om0: 0.3
11    IMAGE:
12        PARAMETERS:
13            exposure_time: 565.0
14            numPix: 101
15            pixel_scale: 0.101
16            psf_type: 'GAUSSIAN'
17            read_noise: 4.2
18            ccd_gain: 3.1
19    SURVEY:
20        PARAMETERS:
21            BANDS: VIS
22            seeing: 0.16
23            magnitude_zero_point: 24.0
24            sky_brightness: 22.35
25            num_exposures: 4
26    SPECIES:
27        GALAXY_1:
28            NAME: LENS
29            LIGHT_PROFILE_1:
30                NAME: SERSIC_ELLIPSE
31                PARAMETERS:
32                    magnitude: 100
33                    center_x: 0.0
34                    center_y: 0.0
35                    R_sersic: 1.0
36                    n_sersic: 1.0
```

```
37         e1: 0
38         e2: 0
39     MASS_PROFILE_1:
40         NAME: SIE
41         PARAMETERS:
42             sigma_v: 250
43             e1: 0.4
44             e2: -0.3
45             center_x: 0.0
46             center_y: 0.0
47 GALAXY_2:
48     NAME: SOURCE
49     LIGHT_PROFILE_1:
50         NAME: SERSIC_ELLIPSE
51         PARAMETERS:
52             center_x: 0
53             center_y: 0
54             magnitude: 10
55             R_sersic: 1
56             n_sersic: 0.01
57             e1: 0.11
58             e2: 0.54
59     MASS_PROFILE_1:
60         NAME: SIE
61         PARAMETERS:
62             e1: 0
63             e2: 0
64             center_x: 0.0
65             center_y: 0.0
```

```
66     NOISE_1:
67         NAME: POISSON_NOISE
68         PARAMETERS:
69             mean: 0.5
70     GEOMETRY:
71         CONFIGURATION_1:
72             NAME: LENSING
73             FRACTION: 1
74             PLANE_1:
75                 OBJECT_1: LENS
76                 PARAMETERS:
77                     REDSHIFT: 2
78             PLANE_2:
79                 OBJECT_1: SOURCE
80                 PARAMETERS:
81                     REDSHIFT: 4
```

REFERENCIAS BIBLIOGRÁFICAS

- [Abadi *et al.*, 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., y Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- [Birrer y Amara, 2018] Birrer, S. y Amara, A. (2018). lenstronomy: Multi-purpose gravitational lens modelling software package. *Physics of the Dark Universe*, 22:189 – 201.
- [Boylan-Kolchin *et al.*, 2009] Boylan-Kolchin, M., Springel, V., White, S. D. M., Jenkins, A., y Lemson, G. (2009). Resolving cosmic structure formation with the Millennium-II Simulation. *Monthly Notices of the Royal Astronomical Society*, 398(3):1150–1164.
- [Bradski, 2000] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [Cantalupo *et al.*, 2013] Cantalupo, S., Lilly, S. J., y Haehnelt, M. G. (2013). Detection of Dark Galaxies and Circum-galactic Filaments Fluorescently Illuminated by the $z=2.4$ Quasar HE0109-3518. En *American Astronomical Society Meeting Abstracts #221*, volumen 221 de *American Astronomical Society Meeting Abstracts*, p. 245.06.
- [Caon *et al.*, 1993] Caon, N., Capaccioli, M., y D'Onofrio, M. (1993). On the shape of the light profiles of early-type galaxies. *mnras*, 265:1013–1021.
- [Chollet, 2015] Chollet, F. (2015). Keras. <https://keras.io>.
- [de Jong *et al.*, 2013] de Jong, J. T. A., Verdoes Kleijn, G. A., Kuijken, K., Valentijn, E. A., van Waerbeke, L., Hildebrandt, H., Erben, T., Hudelot, P., Schneider, P., Viola, M., Blaeu, M., Brescia, M., Choi, A., Conti, I. F., Covone, G., Dolag, K., Dvornik, A., Grado, A., Hoekstra, H., Huang, Z., Irisarri, N., Kekziora, J., König, C., Napolitano, N. R., Norberg, P., Radovich,

- M., Sikkema, G., Simon, P., Tortora, C., van den Busch, J. L. P., van Uitert, E., Vriend, W.-J., Begeman, K. G., Brescia, V., Boxhoorn, D. R., Buddelmeijer, H., Cacciato, M., Choi, J. H., Hutter, A., Köhlinger, F., Merten, J., Napolitano, M., Nicolaci da Costa, L., Radovich, M., Schneider, P., Sikkema, G., Verdoes Kleijn, G. A., Vriend, W.-J., Vulcani, B., y Wolf, C. (2013). The Kilo-Degree Survey. *The Messenger*, 154:44–48.
- [de Vaucouleurs, 1948] de Vaucouleurs, G. (1948). Recherches sur les Nebuleuses Extragalactiques. *Annales d'Astrophysique*, 11:247.
- [Duc y Bornaud, 2008] Duc, P.-A. y Bornaud, F. (2008). Tidal debris from high-velocity collisions as fake dark galaxies: a numerical model of virgohi 21. *The Astrophysical Journal*, 673(2):787.
- [Einstein, 1916] Einstein, A. (1916). The foundation of the general theory of relativity.
- [Euclid Collaboration et al., 2022] Euclid Collaboration, Bretonnière, H., Huertas-Company, M., Boucaud, A., Lanusse, F., Jullo, E., Merlin, E., Tuccillo, D., Castellano, M., Brinchmann, J., Conselice, C. J., Dole, H., Cabanac, R., Courtois, H. M., Castander, F. J., Duc, P. A., Fosalba, P., Guinet, D., Kruk, S., Kuchner, U., Serrano, S., Soubrie, E., Tramacere, A., Wang, L., Amara, A., Auricchio, N., Bender, R., Bodendorf, C., Bonino, D., Branchini, E., Brau-Nogue, S., Brescia, M., Capobianco, V., Carbone, C., Carretero, J., Cavuoti, S., Cimatti, A., Cledassou, R., Congedo, G., Conversi, L., Copin, Y., Corcione, L., Costille, A., Cropper, M., Da Silva, A., Degaudenzi, H., Douspis, M., Dubath, F., Duncan, C. A. J., Dupac, X., Dusini, S., Farrens, S., Ferriol, S., Frailis, M., Franceschi, E., Fumana, M., Garilli, B., Gillard, W., Gillis, B., Giocoli, C., Grazian, A., Grupp, F., Haugan, S. V. H., Holmes, W., Hormuth, F., Hudelot, P., Jahnke, K., Kermiche, S., Kiessling, A., Kilbinger, M., Kitching, T., Kohley, R., Kümmel, M., Kunz, M., Kurki-Suonio, H., Liori, S., Lilje, P. B., Lloro, I., Maiorano, E., Mansutti, O., Marggraf, O., Markovic, K., Marulli, F., Massey, R., Maurogordato, S., Melchior, M., Meneghetti, M., Meylan, G., Moresco, M., Morin, B., Moscardini, L., Munari, E., Nakajima, R., Niemi, S. M., Padilla, C., Paltani, S., Pasian, F., Pedersen, K., Pettorino, V., Pires, S., Poncet, M., Popa, L., Pozzetti, L., Raison, F., Rebolo, R., Rhodes, J., Roncarelli, M., Rossetti, E., Saglia, R., Schneider, P., Secroun, A., Seidel, G., Sirignano, C., Sirri, G., Stanco, L., Starck, J. L., Tallada-Crespí,

P., Taylor, A. N., Tereno, I., Toledo-Moreo, R., Torradeflot, F., Valentijn, E. A., Valenziano, L., Wang, Y., Welikala, N., Weller, J., Zamorani, G., Zoubian, J., Baldi, M., Bardelli, S., Camera, S., Farinelli, R., Medinaceli, E., Mei, S., Polenta, G., Romelli, E., Tenti, M., Vassallo, T., Zacchei, A., Zucca, E., Baccigalupi, C., Balaguera-Antolínez, A., Biviano, A., Borgani, S., Bozzo, E., Burigana, C., Cappi, A., Carvalho, C. S., Casas, S., Castignani, G., Colodro-Conde, C., Coupon, J., de la Torre, S., Fabricius, M., Farina, M., Ferreira, P. G., Flose-Reimberg, P., Fotopoulou, S., Galeotta, S., Ganga, K., Garcia-Bellido, J., Gaztanaga, E., Gozaliasl, G., Hook, I. M., Joachimi, B., Kansal, V., Kashlinsky, A., Keihanen, E., Kirkpatrick, C. C., Lindholm, V., Mainetti, G., Maino, D., Maoli, R., Martinelli, M., Martinet, N., McCracken, H. J., Metcalf, R. B., Morgante, G., Morisset, N., Nightingale, J., Nucita, A., Patrizzii, L., Potter, D., Renzi, A., Riccio, G., Sánchez, A. G., Sapone, D., Schirmer, M., Schultheis, M., Scottez, V., Sefusatti, E., Teyssier, R., Tutusaus, I., Valiviita, J., Viel, M., Whittaker, L., y Knapen, J. H. (2022). Euclid preparation. XIII. Forecasts for galaxy morphology with the Euclid Survey using deep generative models. *aap*, 657:A90.

[Goodfellow *et al.*, 2016] Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.

[Guo *et al.*, 2011] Guo, Q., White, S., Boylan-Kolchin, M., De Lucia, G., Kauffmann, G., Lemson, G., Li, C., Springel, V., y Weinmann, S. (2011). From dwarf spheroidals to cD galaxies: simulating the galaxy population in a Λ CDM cosmology. *mnras*, 413(1):101–131.

[Hawkins, 1997] Hawkins, M. R. S. (1997). Dark galaxies – the dominant population?

[Hawkins *et al.*, 1997] Hawkins, M. R. S., Clements, D., Fried, J. W., Heavens, A. F., Veron, P., Minty, E. M., y van der Werf, P. (1997). The double quasar Q2138-431: lensing by a dark galaxy? *mnras*, 291(4):811–818.

[He *et al.*, 2015] He, K., Zhang, X., Ren, S., y Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv e-prints*, p. arXiv:1512.03385.

[Jackson *et al.*, 1998] Jackson, N., Helbig, P., Browne, I. W. A., Fassnacht, C. D., Koopmans, L. V. E., Marlow, D. R., y Wilkinson, P. N. (1998). Lensing galaxies: light or dark?

- [Jimenez *et al.*, 1997] Jimenez, R., Heavens, A. F., Hawkins, M. R. S., y Padoan, P. (1997). Dark galaxies, spin bias and gravitational lenses. *mnras*, 292(1):L5–L10.
- [Kelly *et al.*, 2016] Kelly, P. L., Rodney, S. A., Treu, T., Strolger, L.-G., Foley, R. J., Jha, S. W., Selsing, J., Brammer, G., Bradač, M., Cenko, S. B., Graur, O., Filippenko, A. V., Hjorth, J., McCully, C., Molino, A., Nonino, M., Riess, A. G., Schmidt, K. B., Tucker, B., von der Linden, A., Weiner, B. J., y Zitrin, A. (2016). DEJA VU ALL OVER AGAIN: THE REAPPEARANCE OF SUPERNOVA REFSDAL. *The Astrophysical Journal*, 819(1):L8.
- [Kingma y Ba, 2014] Kingma, D. P. y Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Lanusse *et al.*, 2018] Lanusse, F., Ma, Q., Li, N., Collett, T. E., Li, C.-L., Ravanbakhsh, S., Mandelbaum, R., y Póczos, B. (2018). CMU DeepLens: deep learning for automatic image-based galaxy-galaxy strong lens finding. *mnras*, 473(3):3895–3906.
- [Liaudat *et al.*, 2023] Liaudat, T., Starck, J.-L., Kilbinger, M., y Frugier, P.-A. (2023). Point spread function modelling for astronomical telescopes: a review focused on weak gravitational lensing studies. *arXiv e-prints*, p. arXiv:2306.07996.
- [Macciò, 2008] Macciò, A. V. (2008). QSO Strong Gravitational Lensing and the Detection of Dark Halos. En Davies, J. I. y Disney, M. J., editores, *Dark Galaxies and Lost Baryons*, volumen 244, pp. 186–195.
- [Marino *et al.*, 2018] Marino, R. A., Cantalupo, S., Lilly, S. J., Gallego, S. G., Straka, L. A., Borisova, E., Pezzulli, G., Bacon, R., Brinchmann, J., Carollo, C. M., Caruana, J., Conseil, S., Contini, T., Diener, C., Finley, H., Inami, H., Leclercq, F., Muzahid, S., Richard, J., Schaye, J., Wendt, M., y Wisotzki, L. (2018). Dark Galaxy Candidates at Redshift ~ 3.5 Detected with MUSE. *apj*, 859(1):53.
- [Meneghetti *et al.*, 2010] Meneghetti, M., Rasia, E., Merten, J., Bellagamba, F., Ettori, S., Mazzotta, P., Dolag, K., y Marri, S. (2010). Weighing simulated galaxy clusters using lensing and X-ray. *aap*, 514:A93.

- [Metcalf *et al.*, 2019] Metcalf, R. B., Meneghetti, M., Avestruz, C., Bellagamba, F., Bom, C. R., Bertin, E., Cabanac, R., Courbin, F., Davies, A., Decenci re, E., Flamary, R., Gavazzi, R., Geiger, M., Hartley, P., Huertas-Company, M., Jackson, N., Jacobs, C., Jullo, E., Kneib, J. P., Koopmans, L. V. E., Lanusse, F., Li, C. L., Ma, Q., Makler, M., Li, N., Lightman, M., Petrillo, C. E., Serjeant, S., Sch fer, C., Sonnenfeld, A., Tagore, A., Tortora, C., Tuccillo, D., Valen t n, M. B., Velasco-Forero, S., Verdoes Kleijn, G. A., y Vernardos, G. (2019). The strong gravitational lens finding challenge. *aap*, 625:A119.
- [Mitchell, 1997] Mitchell, T. (1997). Machine learning, mcgraw-hill international editions. mcgraw-hill.
- [Morgan *et al.*, 2021] Morgan, R., Nord, B., Birrer, S., Lin, J. Y.-Y., y Poh, J. (2021). deeplens-
tronomy: A dataset simulation package for strong gravitational lensing. *Journal of Open Source Software*, 6(58):2854.
- [Petkova *et al.*, 2014] Petkova, M., Metcalf, R. B., y Giocoli, C. (2014). GLAMER - II. Multiple-plane gravitational lensing. *mnras*, 445(2):1954–1966.
- [Rescigno, 2019] Rescigno, U. (2019). Modeling massive galaxy clusters via strong gravitational lensing: the case for Abell 2163. <http://hdl.handle.net/2434/697126>.
- [Rodriguez-Gomez *et al.*, 2018] Rodriguez-Gomez, V., Snyder, G. F., Lotz, J. M., Nelson, D., Pillepich, A., Springel, V., Genel, S., Weinberger, R., Tacchella, S., Pakmor, R., Torrey, P., Marinacci, F., Vogelsberger, M., Hernquist, L., y Thilker, D. A. (2018). The optical morphologies of galaxies in the IllustrisTNG simulation: a comparison to pan-STARRS observations. *Monthly Notices of the Royal Astronomical Society*, 483(3):4140–4159.
- [Rumelhart D. E. Hinton, 1986] Rumelhart D. E. Hinton, G. E. e. a. (1986). Learning representations by back-propagating errors.
- [Russakovsky *et al.*, 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., y Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

- [Schaefer *et al.*, 2018] Schaefer, C., Geiger, M., Kuntzer, T., y Kneib, J. P. (2018). Deep convolutional neural networks as strong gravitational lens detectors. *aap*, 611:A2.
- [Sérsic, 1963] Sérsic, J. L. (1963). Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Boletín de la Asociacion Argentina de Astronomia La Plata Argentina*, 6:41–43.
- [Simon, 2005] Simon, J. D. (2005). *Dark Matter in Dwarf Galaxies: Observational Tests of the Cold Dark Matter Paradigm on Small Scales*. Tesis doctoral, University of California, Berkeley.
- [Smith, 2023] Smith, J. (2023). K-means clustering algorithm: A comprehensive analysis and applications. *Journal of Machine Learning Research*, 15(3):125–150.
- [Tollerud *et al.*, 2008] Tollerud, E. J., Bullock, J. S., Strigari, L. E., y Willman, B. (2008). Hundreds of milky way satellites? luminosity bias in the satellite luminosity function. *The Astrophysical Journal*, 688(1):277.
- [Trentham *et al.*, 2001] Trentham, N., Moller, O., y Ramirez-Ruiz, E. (2001). Completely dark galaxies: their existence, properties and strategies for finding them. *Monthly Notices of the Royal Astronomical Society*, 322(3):658–668.
- [Virtanen *et al.*, 2020] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., y SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
- [Wong *et al.*, 2019] Wong, K. C., Suyu, S. H., Chen, G. C.-F., Rusu, C. E., Millon, M., Sluse, D., Bonvin, V., Fassnacht, C. D., Taubenberger, S., Auger, M. W., Birrer, S., Chan, J. H. H., Courbin, F., Hilbert, S., Tihhonova, O., Treu, T., Agnello, A., Ding, X., Jee, I., Komatsu, E.,

Shajib, A. J., Sonnenfeld, A., Blandford, R. D., Koopmans, L. V. E., Marshall, P. J., y Meylan, G. (2019). H0licow – XIII. a 2.4 per cent measurement of h_0 from lensed quasars: 5.3σ tension between early- and late-universe probes. *Monthly Notices of the Royal Astronomical Society*, 498(1):1420–1439.

[Wright, 2006] Wright, E. L. (2006). A Cosmology Calculator for the World Wide Web.

[Xu et al., 2023] Xu, J.-L., Zhu, M., Yu, N., Zhang, C.-P., Liu, X.-L., Ai, M., y Jiang, P. (2023). Discovery of an isolated dark dwarf galaxy in the nearby universe. *The Astrophysical Journal Letters*, 944(2):L40.