

UNIVERSIDAD DE VALLADOLID
MÁSTER UNIVERSITARIO
Ingeniería Informática



TRABAJO FIN DE MÁSTER

**Desarrollo de técnicas de visión artificial
basadas en aprendizaje profundo para la
delimitación de bordes de galaxias del
espacio distante**

Realizado por **D. Jesús Fernández Iglesias**



Universidad de Valladolid

30 de junio de 2022

Tutores: D. Benjamín Sahelices Fernández

D. Fernando Buitrago Alonso

Resumen

Asignar un tamaño a las galaxias es una tarea extremadamente compleja debido a la naturaleza difusa y mal definida que tienen estos cuerpos. Además, las variaciones en el tiempo de exposición utilizado para recibir información de las mismas provoca que se capten zonas más o menos débiles, variando así el brillo aparente de estas. Actualmente, el proceso de delimitación de bordes de objetos galácticos, el cual brinda valiosa información acerca de la formación y evolución del espacio distante, se realiza de manera manual, con la consiguiente imposibilidad de estimar los límites de miles de millones de galaxias a las que cada vez se tiene más acceso debido al incesante desarrollo de la tecnología en el ámbito de la astrofísica. La clara enmarcación del problema como un reconocimiento de patrones motiva que en este trabajo se desarrollen técnicas de aprendizaje profundo en el estado de arte que sean capaces de asignar un tamaño a las galaxias mediante la estimación de sus límites. Para ello, se utiliza un conjunto de datos del proyecto CANDELS, el mayor programa espacial llevado a cabo por el telescopio Hubble, y se sigue un enfoque de segmentación semántica, obteniendo unos resultados de gran calidad. Además, mostramos cómo la generación de *astronomic augmentations* mediante el desarrollo de distintas combinaciones de bandas del espectro electromagnético, y su aprovechamiento simultáneo para realizar inferencias, induce a unos resultados más preciso sobre el conjunto de test. Dichos resultados abren la puerta a la exploración de técnicas de generación de nuevas imágenes con más datos que puedan ser captados por los telescopios espaciales con el objetivo de su utilización en masa para superar los resultados que se encuentren en el estado del arte.

Descriptores

tamaño de galaxias, visión artificial, reconocimiento de patrones, aprendizaje profundo, telescopio Hubble, segmentación semántica, *astronomic augmentations*

Abstract

Being able to assign a size to galaxies is an extremely complex task due to the diffuse and fuzzy-defined nature of these bodies. In addition, variations in the exposure time used to receive information from galaxies cause more or less weak areas to be captured, thus varying their apparent brightness. Currently, the process of delimiting the edges of galactic objects, which provides valuable information about the formation and evolution of distant space, is done manually, with the consequent impossibility of estimating the limits of billions of galaxies to which we have more and more access due to the incessant development of technology in the field of astrophysics. The clear framing of the problem as a pattern recognition one prompts this work to develop state-of-the-art deep learning techniques that are able to assign a size to galaxies by estimating their boundaries. To do so, we use a dataset from the CANDELS project, the largest programme ever carried out by the Hubble Space Telescope, and follow a semantic segmentation approach, obtaining high quality results. In addition, we show how the generation of astronomic augmentations by developing different combinations of bands of the electromagnetic spectrum, and their simultaneous exploitation to make inferences, leads to more accurate results on the test set. These results open the door to the exploration of techniques for generating new images with more data that can be captured by space telescopes with the aim of using them in large numbers to surpass the state-of-the-art results.

Keywords

galaxy size, artificial vision, pattern matching, deep learning, Hubble telescope, semantic segmentation, *astronomic augmentations*

Agradecimientos

A Benjamín y a Fernando, por su constante dedicación, por darme la oportunidad de colaborar estrechamente con ellos, por proporcionarme los recursos necesarios para poder haber realizado el trabajo y por educarme e introducirme con cercanía en las buenas praxis del mundo de la investigación académica.

A mis padres y a mi hermano, por inculcarme los valores del sacrificio, el esfuerzo y la humildad que siempre han de primar.

A mi pareja, por apoyarme y aconsejarme en todas las decisiones que tomo, ayudarme a crecer, creer en mi y siempre estar ahí.

Índice general

Índice general	VII
Índice de figuras	IX
Índice de tablas	XII
Siglas	XIV
1. Introducción	1
1.1. Contextualización	1
1.2. Estructura del documento	2
1.3. Materiales	2
1.3.1. <i>Software</i>	2
1.3.2. <i>Hardware</i>	3
2. Planificación	4
2.1. Objetivos	4
2.1.1. Objetivo principal	4
2.1.2. Objetivos secundarios	5
2.2. Alcance	5
2.2.1. Definición de requisitos	5
2.2.2. Estructura de descomposición del trabajo	5
2.3. Plazo	9
2.4. Planes de soporte	9
2.4.1. Recursos	9
2.4.2. Riesgos	9
2.5. Metodología de trabajo	10
3. Redes neuronales artificiales convolucionales	16
3.1. Conceptos generales	18

3.1.1 Evolución	18
3.1.2 Arquitectura y operaciones	19
3.1.2.1 Operación de convolución	20
3.1.2.2 Operación de submuestreo	21
3.1.2.3 Capas densas	22
3.2. Equivarianza e invarianza	23
3.3. Arquitecturas modernas	26
3.3.1 ResNet	26
3.3.2 DenseNet	28
3.3.3. EfficientNet	29
3.3.4. DPN	30
3.4. Segmentación semántica	31
3.4.1 Redes en U	32
3.4.2 Redes en U anidadas	33
3.4.3 Arquitecturas con convoluciones dilatadas separables: DeepLabV3	34
4. Truncaciones astronómicas	38
4.1. Enfoques clásicos	39
4.2. Enfoques modernos	42
5. Metodología experimental	44
5.1. Conjunto de datos	44
5.1.1. Descripción	44
5.1.2. <i>Astronomic Augmentations</i>	45
5.1.3. Exploración	47
5.1.4. Transformación	49
5.1.4.1. Eliminado de galaxias secundarias	50
5.1.4.2. Rellenado de huecos por superposición de cuerpos	51
5.2. Diseño experimental	52
5.2.1. Configuraciones arquitectónicas	52
5.2.2 Hiperparámetros y características del entrenamiento	56
6. Resultados	60
6.1. Evolución del aprendizaje	60
6.2. Resultados sobre el conjunto de test	68
6.3. Análisis del efecto <i>astronomic augmentations</i>	68
6.4. Análisis visual del efecto de las convoluciones agrupadas	74
6.5. Análisis del mejor modelo	75
7. Conclusiones y líneas de trabajo futuras	81
7.1. Conclusiones	81
7.2. Líneas de trabajo futuras	82
Bibliografía	84

Índice de figuras

2.1. Estructura de descomposición del trabajo.	14
2.2. Diagrama de Gantt de las actividades del proyecto.	15
3.1. Extracción de características y aprendizaje implícitos en las aproximaciones <i>deep learning</i>	17
3.2. Diagrama de Venn mostrando la relación entre la inteligencia artificial y algunos de sus campos.	18
3.3. Esquema de la arquitectura de la red LeNet-5.	19
3.4. Efecto visual de la generación de mapas de características distintos en función del filtro que se aplique.	21
3.5. Efecto visual de la aplicación de <i>pooling</i> del máximo (izquierda), promedio (centro) y mínimo (derecha).	22
3.6. Equivarianza a la traslación de la operación de convolución.	23
3.7. Invarianza a la traslación.	24
3.8. Invarianza a la rotación.	25
3.9. Funciones anidadas, no anidadas y función ideal.	26
3.10. Bloque normal (izquierda) frente a bloque residual (derecha).	27
3.11. Bloque ResNet (izquierda) frente a bloque DenseNet (derecha).	28
3.12. Escalados en anchura, profundidad, resolución y compuesto.	29
3.13. Representaciones de arquitecturas ResNet, DenseNet y de ruta dual como redes neuronales recurrentes de alto nivel.	31
3.14. Algunas de las categorías de problemas más comunes en visión artificial.	32
3.15. Arquitectura de una red en U.	33
3.16. Arquitectura de una U-Net++.	34
3.17. Conexión transversal rediseñada en una U-Net++.	34
3.18. Convoluciones dilatadas con diferentes radios y <i>kernel 3x3</i>	35
3.19. <i>Atrous Spatial Pyramid Pooling</i>	35
3.20. Arquitectura DeepLabV3.	37
4.1. Tamaño de las galaxias en función del tiempo de exposición.	38
4.2. Cálculo de isofotas en una galaxia simulada.	39
4.3. Luminosidad y brillo aparente de un cuerpo.	40

4.4.	<i>Half-light radius</i> .	41
4.5.	Imágenes de galaxias que muestran la ubicación de sus r_e y $R1$ (extraído de [33]).	43
4.6.	Imágenes de galaxias que muestran la ubicación de sus r_e y R_{edge} (extraído de [2]).	43
5.1.	Ejemplo de imagen de una galaxia junto con su máscara asociada.	45
5.2.	Taxonomía de <i>data augmentations</i> aplicadas en visión por computador.	46
5.3.	Combinación de las bandas H, J, I y V para formar las <i>astronomic augmentations</i> .	47
5.4.	Proporción de imágenes según campo de exposición.	48
5.5.	Gráfico <i>ridgeline</i> mostrando la distribución del tamaño de las galaxias según el campo de exposición.	49
5.6.	Gráfico de líneas mostrando la relación entre el número de imágenes y el tamaño de la galaxia de cada una, ordenadas de menor a mayor superficie, según el campo de exposición.	50
5.7.	Galaxia más masiva y galaxias más pequeñas del conjunto de datos.	50
5.8.	Etapas inicial del <i>pipeline</i> de transformación.	51
5.9.	Etapas secundaria del <i>pipeline</i> de transformación.	51
5.10.	Número de parámetros de cada combinación arquitectura de segmentación - <i>encoder</i> para el caso del experimento base.	53
5.11.	Convoluciones agrupadas utilizadas en AlexNet (extraído de [19]).	54
5.12.	Bloque ResNet (izquierda) frente a bloque ResNext (derecha).	55
5.13.	Arquitectura diseñada para los experimentos con <i>astronomic augmentations</i> utilizando como red de segmentación una <i>U-Net ++</i> .	56
5.14.	<i>Data augmentation</i> aplicado a la muestra de entrenamiento.	58
5.15.	Comportamiento del índice Sørensen–Dice frente a la tasa de acierto, el <i>precision</i> y el <i>recall</i> .	59
6.1.	Evolución del <i>loss</i> sobre los conjuntos de entrenamiento y validación del experimento base para los distintos <i>encoders</i> que se incrustan en una arquitectura de tipo <i>U-Net</i> .	61
6.2.	Evolución del <i>loss</i> sobre los conjuntos de entrenamiento y validación del experimento visual para los distintos <i>encoders</i> que se incrustan en una arquitectura de tipo <i>U-Net</i> .	62
6.3.	Evolución del <i>loss</i> sobre los conjuntos de entrenamiento y validación del experimento base para los distintos <i>encoders</i> que se incrustan en una arquitectura de tipo <i>U-Net ++</i> .	64
6.4.	Evolución del <i>loss</i> sobre los conjuntos de entrenamiento y validación del experimento visual para los distintos <i>encoders</i> que se incrustan en una arquitectura de tipo <i>U-Net ++</i> .	65
6.5.	Evolución del <i>loss</i> sobre los conjuntos de entrenamiento y validación del experimento base para los distintos <i>encoders</i> que se incrustan en una arquitectura de tipo DeepLabV3.	66
6.6.	Evolución del <i>loss</i> sobre los conjuntos de entrenamiento y validación del experimento visual para los distintos <i>encoders</i> que se incrustan en una arquitectura de tipo DeepLabV3.	67

6.7. Gráfico de barras del coeficiente de Sørensen–Dice medio según arquitectura de segmentación y conjunto de datos.	70
6.8. Gráfico de barras del coeficiente de Sørensen–Dice medio según <i>encoder</i> y conjunto de datos.	72
6.9. Mapas de características obtenidos tras la aplicación de las convoluciones agrupadas a la galaxia DISKIRR-4532-cosmos.	75
6.10. Mapas de características obtenidos tras la aplicación de las convoluciones agrupadas a la galaxia DISKSPH-14886-goodss.	76
6.11. Mapas de características obtenidos tras la aplicación de las convoluciones agrupadas a la galaxia DISKSPH-18389-goodsn.	77
6.12. 10 mejores galaxias delimitadas por el mejor modelo. Para cada galaxia se tiene, de izquierda a derecha y de arriba a abajo, borde real, borde predicho, comparación de bordes, máscara real, máscara predicha y diferencia de máscaras.	79
6.13. 10 peores galaxias delimitadas por el mejor modelo. Para cada galaxia se tiene, de izquierda a derecha y de arriba a abajo, borde real, borde predicho, comparación de bordes, máscara real, máscara predicha y diferencia de máscaras.	80

Índice de tablas

2.1. Requisito <i>RG-1</i> del proyecto.	6
2.2. Requisito <i>RG-2</i> del proyecto.	6
2.3. Requisito <i>RG-3</i> del proyecto.	6
2.4. Requisito <i>RG-4</i> del proyecto.	7
2.5. Requisito <i>RG-5</i> del proyecto.	7
2.6. Requisito <i>RG-6</i> del proyecto.	7
2.7. Requisito <i>RG-7</i> del proyecto.	8
2.8. Requisito <i>RG-8</i> del proyecto.	8
2.9. Requisito <i>RG-9</i> del proyecto.	8
2.10. Riesgo <i>Ri-01</i> del proyecto.	10
2.11. Riesgo <i>Ri-02</i> del proyecto.	10
2.12. Riesgo <i>Ri-03</i> del proyecto.	10
2.13. Riesgo <i>Ri-04</i> del proyecto.	11
2.14. Riesgo <i>Ri-05</i> del proyecto.	11
2.15. Riesgo <i>Ri-06</i> del proyecto.	11
2.16. Riesgo <i>Ri-07</i> del proyecto.	12
2.17. Riesgo <i>Ri-08</i> del proyecto.	12
2.18. Riesgo <i>Ri-09</i> del proyecto.	12
2.19. <i>Sprint goals</i> con sus correspondientes fechas de inicio, fin y las horas dedicadas.	13
5.1. Estadísticos descriptivos de la superficie de las galaxias, en píxeles, según el campo de exposición.	49
6.1. Coeficiente de Sørensen–Dice medio de cada experimento sobre el conjunto de test.	69
6.2. Coeficiente de Sørensen–Dice medio de cada arquitectura de segmentación según el conjunto de datos utilizado.	70
6.3. Coeficiente de Sørensen–Dice medio de cada <i>encoder</i> según el conjunto de datos utilizado.	71
6.4. <i>Precision</i> , <i>recall</i> y coeficiente de Sørensen–Dice medios sobre el conjunto de test según el campo de exposición y su prevalencia.	77

Siglas

ACS *Advanced Camera for Surveys*

ADAM *Adaptive Moment Estimation*

ANN *Artificial Neural Network*

API *Application Programming Interface*

ASPP *Atrous Spatial Pyramid Pooling*

CANDELS *Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey*

CNN *Convolutional Neural Network*

CPU *Central Processing Unit*

COSMOS *Cosmological Evolution Survey*

DPN *Dual Path Networks*

EDT *Estructura de Descomposición del Trabajo*

EGS *Extended Groth Strip*

FC *Fully Connected*

FITS *Flexible Image Transport System*

GIR *Grupo de Investigación Reconocido*

GOODS-N *Great Observatories Origins Deep Survey - North*

GOODS-S *Great Observatories Origins Deep Survey - South*

GPU *Graphics Processing Unit*

GTC *Gran Telescopio de Canarias*

HORNN *High Order Recurrent Neural Networks*

HST *Hubble Space Telescope*

ILSVRC *ImageNet Large Scale Visual Recognition Challenge*

LSB *Low Surface Brightness*

MoBiVAP *Modelado, Biomecánica y Visualización Avanzada del Patrimonio*

NLP *Natural Language Processing*

OS *Operative System*

PMBOK *Project Management Body of Knowledge*

RAM *Random Access Memory*

UDS *Ultra Deep Survey*

VGG *Visual Geometry Group*

WFC3 *Wide Field Camera 3*

1: Introducción

Este primer capítulo del documento tiene el objetivo de contextualizar el problema a abordar, resumir los contenidos que se tratan en cada capítulo y enumerar la relación de materiales utilizados que han permitido el desarrollo del trabajo.

1.1. Contextualización

El incipiente desarrollo tecnológico que está experimentando la astrofísica requiere del desarrollo de soluciones que permitan mantener de manera sostenida el ritmo de crecimiento y aprovechar la nueva información disponible. Sin ir más lejos, el espectrógrafo de nueva generación desplegado en el telescopio James Webb, el cual pretende sustituir a sus homólogos Hubble y Spitzer, es capaz de observar 100 galaxias de manera simultánea. La complejidad y masividad de la información a la que cada vez se tiene más acceso necesita un desarrollo paralelo de técnicas de gestión y tratamiento de la información en el estado del arte que sean capaces de extraer conocimiento aplicable que ejerza como catalizador para la obtención de nueva información aún más valiosa.

Uno de los problemas más interesantes de abordar en el contexto de la astronomía es el estudio del tamaño de las galaxias. Ser capaces de delimitar con precisión sus cuerpos nos otorga información acerca de los procesos físicos que están aconteciendo en ellas, información que es utilizada para completar los numerosos cabos sueltos presentes en las teorías de la evolución y la formación del universo distante. El proceso de delimitación de los cuerpos galácticos es realizado a mano, con la consiguiente lentitud e imposibilidad de aplicarlo a las miles de millones de galaxias que los cada vez más avanzados dispositivos de captura instalados en los telescopios nos permiten obtener. Por ello, se torna una labor fundamental la elaboración de procedimientos de corte informático que permitan extraer los bordes de las galaxias en un entorno automático, veloz y escalable.

El cálculo del tamaño de las galaxias en el espacio distante representa, de manera palmaria, un problema de reconocimiento de patrones, por lo que el paso natural para

abordar la tarea desde una perspectiva automatizada es el de desarrollar algorítmica perteneciente al campo de conocimiento de la visión artificial. La tecnología emergente, y ya líder, en este campo es toda aquella relacionada con el aprendizaje automático y, más concretamente, con el aprendizaje profundo. La excelente capacidad que tienen algunos procedimientos *deep learning* para extraer patrones valiosos en conjuntos de información desestructurada y con relaciones espaciales intrínsecas provoca que este trabajo se centre en el desarrollo y aplicación de dicha tecnología. Los datos utilizados provienen del proyecto [CANDELS](#), el proyecto de mayor magnitud llevado a cabo por el telescopio Hubble.

1.2. Estructura del documento

En este primer capítulo del trabajo se realiza una contextualización y justificación del interés del problema a tratar, además de presentar brevemente la relación de contenidos tratados en el documento y de enumerar la lista de materiales utilizados para la conducción de la investigación. El segundo capítulo describe la planificación y metodología de trabajo seguidas en el proyecto. El tercer capítulo introduce conceptos de aprendizaje profundo que cobrarán vital importancia durante el diseño y ejecución de la experimentación realizada. El cuarto capítulo introduce el concepto de las truncaciones astronómicas y las distintas aproximaciones no automatizadas que se han ido utilizando a lo largo de los años para determinar y delimitar los bordes de las galaxias. El quinto capítulo explica la relación de métodos experimentales de aprendizaje profundo diseñados y puestos en práctica, cuyos resultados serán mostrados y analizados en el sexto capítulo. Un último capítulo será el encargado de albergar las conclusiones del trabajo y de sugerir líneas futuras de trabajo.

1.3. Materiales

La relación de materiales que ha permitido el desarrollo del proyecto puede subdividirse en 2 categorías, *software* y *hardware*.

1.3.1. *Software*

El *software* que se ha utilizado como apoyo a la conducción de la experimentación realizada, y el papel ejercido, es el siguiente:

- **Python:** como lenguaje de programación generalista en el que se han codificado todos los experimentos. Las bibliotecas *software* más importantes utilizadas han sido:
 - **PyTorch:** como biblioteca de aprendizaje automático en la que se han programado los modelos *deep learning* así como las utilidades necesarias para ejercer los procesos de aprendizaje.
 - **OpenCV:** como biblioteca de visión artificial en la que se han programado rutinas de transformación y tratamiento de las imágenes.

- **Seaborn**: como biblioteca de generación de gráficos.
- **NumPy**: como biblioteca de cálculo numérico y análisis de datos de propósito generalista.
- **pickle**: como biblioteca de serialización y deserialización de objetos para su almacenamiento persistente en memoria.
- **PIL**: como biblioteca de tratamiento de imágenes de propósito generalista.
- **R**: como lenguaje de programación en el que se han realizado tareas de análisis de datos y de generación de representaciones gráficas. Las bibliotecas *software* más importantes utilizadas han sido:
 - **ggplot2**: como biblioteca de generación de gráficos.
 - **ggridges**: como biblioteca de generación de gráficos.
- **CometML**: como plataforma *cloud* de *tracking* y monitorización en tiempo real de experimentos.
- **CUDA**: en su versión 11.6, como *software* para la computación en la tarjeta gráfica. Gestionado a través de PyTorch.
- **Shell scripting**: como medio para el lanzamiento y puesta en cola de los experimentos.

1.3.2. Hardware

Para soportar toda la experimentación realizada se ha utilizado un equipo informático cuyas características se detallan a continuación:

- **OS**: Ubuntu 20.04.4 LTS.
- **Kernel** de linux: 5.13.0-48-generic.
- **CPU**: AMD Ryzen 9 5900X.
- **Cores**: 1 *socket*, con 12 *cores* por *socket* y 2 hilos por *core*.
- **RAM**: 64 GigaBytes.
- **GPU**: NVIDIA RTX A5000.
- **RAM GPU**: 24 GigaBytes.
- Capacidad disco duro: 6 TeraBytes.

2: Planificación

Este capítulo está orientado a ofrecer la planificación que marca el trabajo a realizar en el presente proyecto de fin de máster. La construcción y presentación de los artefactos se ha estructurado de acuerdo al *Project Management Body of Knowledge (PMBOK)* [1], el cuál incorpora buenas prácticas para la planificación y gestión de cualquier tipo de proyectos, como por ejemplo los que se encuentran dentro del mundo de la investigación académica, nicho al que pertenece el que se va a planificar a continuación. Primero se hará una revisión de los objetivos del trabajo, siendo posteriormente especificados los requisitos del proyecto y las actividades necesarias para alcanzar los objetivos con su correspondiente cronograma asociado. Por último, se hablará de los recursos necesarios para completar el proyecto, los riesgos del mismo y la metodología de trabajo seguido.

2.1. Objetivos

Los objetivos del trabajo que se definen en esta sección tratan de procedimentar todas las acciones a realizar durante el ciclo de vida del proyecto y sirven como elemento de soporte a la toma de decisiones.

2.1.1. Objetivo principal

El objetivo prioritario del presente trabajo es el de desarrollar técnicas avanzadas de aprendizaje profundo que permitan detectar bordes de galaxias con una precisión razonable. Para cuantificar y ser capaces de medir el objetivo, se pretenden lograr predicciones que, en término medio y comparándolas con los bordes reales, obtengan al menos un 80 % de coeficiente Sørensen-Dice. Parte de la dificultad residirá en que las galaxias con las que se trabajará tienen la característica de presentar un bajo brillo superficial y de encontrarse en el espacio distante, condición que confiere a la información capturada de las mismas una gran cantidad de contaminación y ruido.

2.1.2. Objetivos secundarios

La relación de objetivos secundarios del trabajo se expone a continuación:

- Asimilar conceptos básicos del ámbito de la astronomía y la astrofísica que permitan trabajar con soltura y comodidad con datos provenientes de telescopios espaciales.
- Aprender y reforzar la integración de *software* astroinformático como parte del proceso de análisis, visualización y tratamiento de información espacial.
- Dominar técnicas clásicas de visión por computador como la detección de regiones convexas.
- Desarrollar conceptos en el estado del arte del ámbito del aprendizaje profundo, concretamente del paradigma de las redes neuronales convolucionales.
- Comprender, tratar y modificar conjuntos de datos astronómicos procedentes del telescopio Hubble que se encuentran en el estado del arte para poder maximizar el posterior rendimiento de las tecnologías basadas en aprendizaje profundo.

2.2. Alcance

El alcance puede definirse como el trabajo que debe realizarse para lograr los objetivos de un proyecto [1], por lo que establece sus límites. De esta manera, los procesos que componen el alcance tratan de restringir las actividades necesarias para lograr la finalización satisfactoria del proyecto. La definición de requisitos y la estructura de descomposición del trabajo son dos de los procesos más importantes que ayudan a cimentar la definición del alcance.

2.2.1. Definición de requisitos

Los requisitos del proyecto han sido derivados de las necesidades del trabajo a realizar, siendo condicionados por las oportunidades y amenazas con capacidad de materializarse durante su realización. Los requisitos más importantes identificados en el proyecto pueden apreciarse en las Tablas [2.1](#), [2.2](#), [2.3](#), [2.4](#), [2.5](#), [2.6](#), [2.7](#), [2.8](#) y [2.9](#).

2.2.2. Estructura de descomposición del trabajo

En esta sección se creará la Estructura de Descomposición del Trabajo ([EDT](#)) del proyecto. Se realizará un desglose jerárquico de las actividades necesarias para cumplir con éxito los objetivos del proyecto. Como consecuencia de la íntima relación que guarda el presente trabajo con la estancia en el grupo de investigación que le ha precedido algunas de las actividades realizadas en el seno del grupo también serán incluidas como parte del trabajo presentado en este documento. Su mutua dependencia y su carácter de realización cíclico permite tratarlos como único ente. Por ello, tanto en el [EDT](#) como en el posterior

ID	<i>RG-1</i>
Nombre	Inicio del trabajo.
Descripción	El trabajo comenzará el día 1 de junio de 2022
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Tiempo
Prioridad	Alta
Partes interesadas	Alumno, tutores

Tabla 2.1: Requisito *RG-1* del proyecto.

ID	<i>RG-2</i>
Nombre	Finalización del trabajo.
Descripción	El trabajo finalizará el día 29 de junio de 2022
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Tiempo
Prioridad	Alta
Partes interesadas	Alumno, tutores

Tabla 2.2: Requisito *RG-2* del proyecto.

ID	<i>RG-3</i>
Nombre	Temática del trabajo.
Descripción	La temática del trabajo pivotará entorno a la aplicación de modelos <i>deep learning</i> para la detección de bordes en galaxias de bajo brillo superficial en el espacio distante
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Alcance
Prioridad	Alta
Partes interesadas	Alumno, tutores

Tabla 2.3: Requisito *RG-3* del proyecto.

ID	<i>RG-4</i>
Nombre	Conjunto de datos.
Descripción	Los <i>inputs</i> con los que se alimentarán los modelos <i>deep learning</i> desarrollados será un conjunto de datos en el estado del arte construido a partir de información suministrada por el telescopio Hubble
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Alcance
Prioridad	Alta
Partes interesadas	Alumno, tutores

Tabla 2.4: Requisito *RG-4* del proyecto.

ID	<i>RG-5</i>
Nombre	Biblioteca de aprendizaje automático.
Descripción	Se utilizará la biblioteca <i>software</i> PyTorch, desde su API en Python, como herramienta para la conducción de la experimentación <i>deep learning</i> .
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Alcance
Prioridad	Media
Partes interesadas	Alumno

Tabla 2.5: Requisito *RG-5* del proyecto.

ID	<i>RG-6</i>
Nombre	Biblioteca de visión artificial.
Descripción	Se utilizará la biblioteca <i>software</i> OpenCV, desde su API en Python, como herramienta para la conducción de los procesos auxiliares de visión artificial
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Alcance
Prioridad	Media
Partes interesadas	Alumno

Tabla 2.6: Requisito *RG-6* del proyecto.

ID	<i>RG-7</i>
Nombre	<i>Astronomic augmentations.</i>
Descripción	Se utilizará la información autocontenida en ficheros FITS astronómicos para la generación de distintas vistas de una misma galaxia y, posteriormente, poder utilizar toda dicha información para maximizar el rendimiento que se obtenga de los algoritmos <i>deep learning</i>
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Alcance
Prioridad	Alta
Partes interesadas	Alumno, tutores

Tabla 2.7: Requisito *RG-7* del proyecto.

ID	<i>RG-8</i>
Nombre	Sucesión a la estancia en el GIR MoBiVAP .
Descripción	El trabajo realizado en este proyecto recogerá, estudiará y extenderá los resultados obtenidos en la estancia en el grupo de investigación MoBiVAP .
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Alcance
Prioridad	Alta
Partes interesadas	Alumno, tutores, GIR

Tabla 2.8: Requisito *RG-8* del proyecto.

ID	<i>RG-9</i>
Nombre	Carga de trabajo
Descripción	El trabajo realizado en este proyecto deberá ocupar 150 horas.
Tipo	Gestión del proyecto
Alcance, tiempo o coste	Tiempo
Prioridad	Alta
Partes interesadas	Alumno

Tabla 2.9: Requisito *RG-9* del proyecto.

cronograma se mostrarán las actividades conjuntas relativas a ambos periodos, con una estimación temporal proporcional a la cantidad de trabajo realizada en el presente proyecto de fin de máster. La EDT se encuentra en la Figura 2.1.

2.3. Plazo

Con el objetivo de estimar la fecha de finalización y duración del proyecto, conocer las fechas en las que se llevan a cabo las actividades e identificar las posibles dependencias entre ellas se desarrolla el diagrama de Gantt de la Figura 2.2. La duración total del proyecto se estima en 150 horas de trabajo, repartidas a lo largo de 4 semanas (comenzando el 2 de junio y finalizando el 30 de junio). Las actividades se han desglosado de acuerdo a lo mostrado en la estructura de descomposición del trabajo.

2.4. Planes de soporte

Como parte de los planes de soporte se hará un desglose de la gestión de los recursos y de la gestión de los riesgos del proyecto.

2.4.1. Recursos

La gestión de los recursos define cómo estimar, adquirir, gestionar y utilizar los recursos físicos. El presente proyecto únicamente necesita de un recurso físico para poder ser realizado, una tarjeta gráfica en la que se realizarán los cómputos necesarios para ajustar los modelos de aprendizaje profundo, soportada y gobernada por un servidor de especificaciones medias. Se estima como necesaria una tarjeta gráfica con una memoria RAM mínima de 8 GigaBytes. Dicho requisito queda completamente cubierto por el *hardware* suministrado por los profesores Benjamín Sahelices Fernández y Fernando Buitrago Alonso, concretamente una GPU Nvidia RTX A5000 con 24 GigaBytes de memoria. El uso de dicho componente será intensivo durante la etapa de experimentación, concretamente en la fase de aprendizaje de los modelos de aprendizaje profundo.

2.4.2. Riesgos

Una de las partes más importantes a realizar durante toda etapa de planificación de un proyecto es la evaluación de riesgos. La identificación de riesgos del proyecto se encuentra en las Tablas 2.10, 2.11, 2.12, 2.13, 2.14, 2.15, 2.16, 2.17 y 2.18. Para la identificación y modo de valoración de los riesgos se utiliza la revisión de métodos para la gestión de riesgos de [26].

Los 3 riesgos identificados como más peligrosos son, por orden ascendente, la presencia de un conjunto de datos con excesivo ruido y/o contaminación, la estimación optimista

ID	<i>Ri-01</i>
Descripción	Enfermedad que impida el correcto desarrollo del trabajo.
Probabilidad	1
Impacto	10
<i>Score</i>	10
Estrategia de prevención	Seguir los consejos e salud promovidos por las autoridades sanitarias.

Tabla 2.10: Riesgo *Ri-01* del proyecto.

ID	<i>Ri-02</i>
Descripción	Mala comprensión inicial del proyecto.
Probabilidad	1
Impacto	9
<i>Score</i>	9
Estrategia de prevención	Mantener una reunión de planificación inicial con los tutores donde se expliquen los resultados esperados.

Tabla 2.11: Riesgo *Ri-02* del proyecto.

ID	<i>Ri-03</i>
Descripción	Falta de bibliografía adecuada.
Probabilidad	1
Impacto	8
<i>Score</i>	8
Estrategia de prevención	Consultar a los tutores por lugares donde obtener material adecuado.

Tabla 2.12: Riesgo *Ri-03* del proyecto.

del tiempo requerido para realizar el diseño, ejecución y validación de la experimentación y la estimación optimista del tiempo requerido para documentar el trabajo.

2.5. Metodología de trabajo

La metodología que se ha seguido durante el desarrollo de la estancia en el grupo de investigación ha sido *Scrum Of REsearch (SCORE)* [14]. Esta metodología, que pertenece al marco de los desarrollos ágiles, es ideal para lidiar con la incertidumbre y

ID	<i>Ri-04</i>
Descripción	<i>Dataset</i> con excesivo ruido y/o contaminación.
Probabilidad	6
Impacto	7
<i>Score</i>	42
Estrategia de prevención	Tratar el conjunto de datos para mitigar las fuentes de contaminación y desarrollar técnicas de aprendizaje profundo robustas.

Tabla 2.13: Riesgo *Ri-04* del proyecto.

ID	<i>Ri-05</i>
Descripción	Incapacidad de creación de modelos que consigan resultados de calidad.
Probabilidad	3
Impacto	8
<i>Score</i>	24
Estrategia de prevención	Desarrollar modelos cuya complejidad sea proporcional a la del problema a abordar.

Tabla 2.14: Riesgo *Ri-05* del proyecto.

ID	<i>Ri-06</i>
Descripción	Falta de recursos <i>hardware</i> para ajustar en un tiempo razonable modelos <i>deep learning</i> de gran tamaño.
Probabilidad	2
Impacto	6
<i>Score</i>	12
Estrategia de prevención	Planificar la adquisición de recursos <i>hardware</i> suficientemente potentes.

Tabla 2.15: Riesgo *Ri-06* del proyecto.

ID	<i>Ri-07</i>
Descripción	Estimación optimista del tiempo requerido para realizar el diseño, ejecución y validación de la experimentación
Probabilidad	8
Impacto	8
<i>Score</i>	64
Estrategia de prevención	Tener presente el calendario en todo momento y adaptar dinámicamente la exigencia de la experimentación al mismo.

Tabla 2.16: Riesgo *Ri-07* del proyecto.

ID	<i>Ri-08</i>
Descripción	Estimación optimista del tiempo requerido para documentar el trabajo.
Probabilidad	8
Impacto	8
<i>Score</i>	64
Estrategia de prevención	Tener presente el calendario en todo momento y adaptar la extensión y los contenidos de la memoria.

Tabla 2.17: Riesgo *Ri-08* del proyecto.

ID	<i>Ri-09</i>
Descripción	Finalización tardía del proyecto .
Probabilidad	4
Impacto	10
<i>Score</i>	40
Estrategia de prevención	Tener presente el calendario en todo momento y reservar días flotador ante la ocurrencia de posibles imprevistos.

Tabla 2.18: Riesgo *Ri-09* del proyecto.

ID	<i>Sprint goal</i>	Fecha de inicio	Fecha de fin	Horas
SG-1	Estudio de bibliografía	02/06/2022	03/06/22	10h
SG-2	Exploración y transformación del conjunto de datos	03/06/2022	04/06/22	10h
SG-3	Diseño y ejecución de la experimentación	04/06/2022	11/06/22	35h
SG-4	Validación y análisis de resultados	11/06/2022	17/06/22	30h

Tabla 2.19: *Sprint goals* con sus correspondientes fechas de inicio, fin y las horas dedicadas.

constante evolución y adaptación inherente en dominios de investigación. SCORE puede ser visto como una especialización del conocido *scrum* especialmente diseñado para las actividades académicas e investigadoras. Tal y como marca la metodología, la frecuencia de las reuniones y la intensidad en el proceso de experimentación y documentación han sido adaptados a las exigencias del trabajo. Concretamente, han sido definidos 4 *sprints*, cada uno con duración variable, conformando así algo más de la mitad de las 4 semanas disponibles entre el inicio y el fin de las actividades enmarcadas en el proyecto. Los *sprint goals*, con sus correspondientes fechas asociadas, se recogen en la Tabla 2.19. Las actividad correspondiente con la redacción de la presente memoria no fue incluida en los *sprints goals* al considerarse un trabajo transversal a todo el proyecto, donde se ha de recoger toda la información recabada durante su realización.

Un aspecto importante a recalcar es que el presente trabajo de fin de máster ha sido realizado tras la estancia en el grupo de investigación MoBiVAP. Las tareas realizadas en el marco de la estancia guardan íntima relación con las expuestas en el presente documento, siendo indivisibles y sirviendo como trabajo de campo e investigación preliminar, por lo que los resultados y experimentos mostrados en esta memoria son el producto de la recopilación de las tareas de ambos periodos.

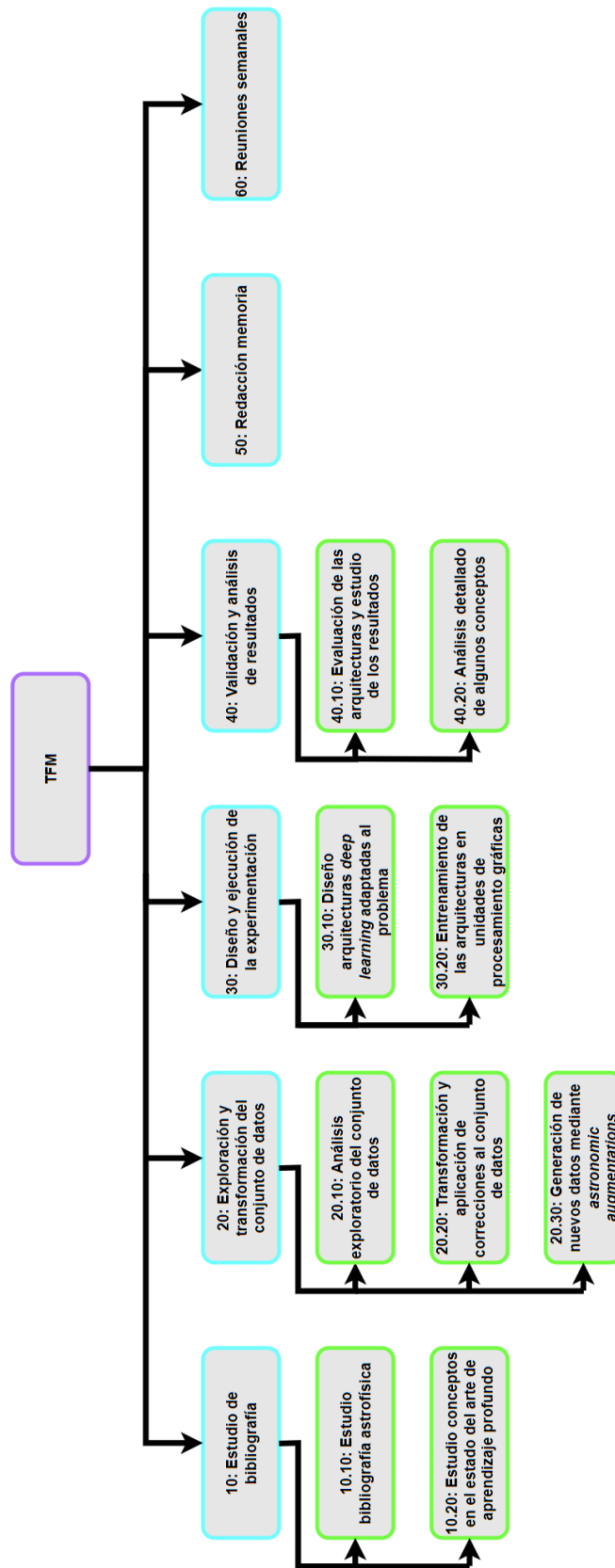


Figura 2.1: Estructura de descomposición del trabajo.

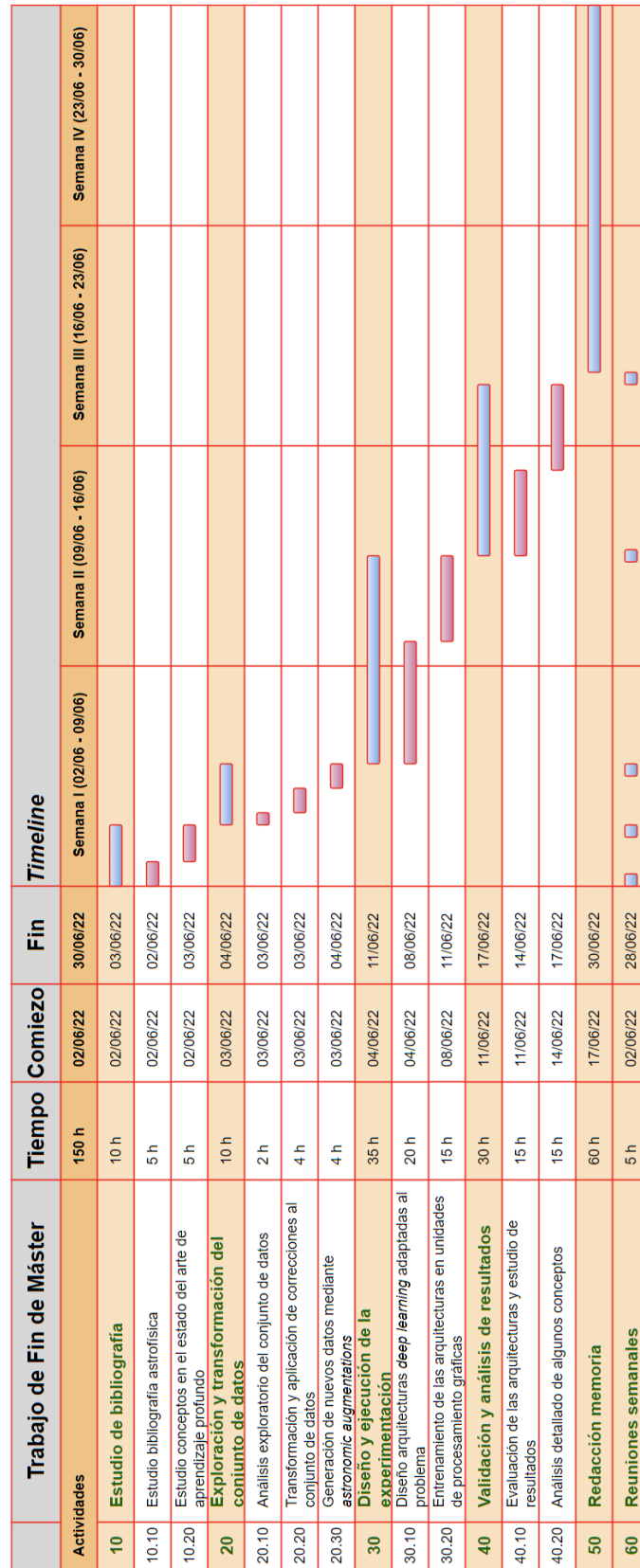


Figura 2.2: Diagrama de Gantt de las actividades del proyecto.

3: Redes neuronales artificiales convolucionales

Desde que el término inteligencia artificial fuese introducido por primera en el año 1956 por John McCarthy en la Conferencia de Dartmouth, considerado como el evento germen de la inteligencia artificial como esfera o campo de actividad, se han ofrecido multitud de definiciones para intentar resumir lo que este término significa. En este trabajo se ha optado por elegir la definición dada por dicho autor en [22], definiendo la inteligencia artificial como la ciencia y la ingeniería de la fabricación de máquinas inteligentes, especialmente de programas informáticos inteligentes. Está relacionada con la tarea similar de utilizar los ordenadores para comprender la inteligencia humana, pero la inteligencia artificial no tiene que limitarse a métodos biológicamente observables.

Uno de los campos del conocimiento que residen bajo el abanico de la inteligencia artificial es el aprendizaje automático, más comúnmente conocido por el término inglés *machine learning*. Antiguamente, una de las definiciones más comunes era la ofrecida por Arthur Samuel: “El campo de estudio que da a los ordenadores la capacidad de aprender sin ser programados explícitamente para ello”. Actualmente, dicha definición se ha quedado algo anticuada, prefiriéndose la ofrecida por Tom Mitchell: “Se dice que un programa de ordenador aprende de la experiencia E con respecto a una serie de tareas T y una medida de rendimiento P , si su rendimiento en las tareas de T , medido por P , mejora con la experiencia E ” [23].

La amplia mayoría de los problemas pertenecientes al aprendizaje automático pueden ser asignados o bien al aprendizaje supervisado, donde se dispone de un conjunto de datos y ya se conoce cómo debe ser la salida correcta de los algoritmos, teniendo la idea de que hay una relación entre la entrada de los mismos y su salida, o al aprendizaje no supervisado, abordando los problemas en los que se dispone de poca o ninguna idea de cómo deberían ser los resultados, tratando de obtener una estructura a partir de datos en los que no se conoce necesariamente el efecto de las variables. Una de las familias

de modelos pertenecientes al aprendizaje automático y que abarcan tanto la perspectiva supervisada como la no supervisada son las *Artificial Neural Networks* (ANNs).

Las redes neuronales artificiales pueden ser definidas como un sistema de aprendizaje computacional que trata de traducir y transformar una entrada de datos en una salida deseada. El concepto de red neuronal se inspiró en la biología humana y en el modo en que las neuronas del cerebro humano funcionan en armonía para comprender las percepciones captadas por los sentidos. La primera red neuronal artificial fue diseñada en el año 1943 por Warren McCulloch y Walter Pitts. En la actualidad, los tipos y familias de redes neuronales existentes conforman un árbol muy extenso, centrándose este trabajo en las redes neuronales convolucionales profundas y en las redes de segmentación semántica, las cuales serán explicadas con detalle en estadios posteriores del documento.

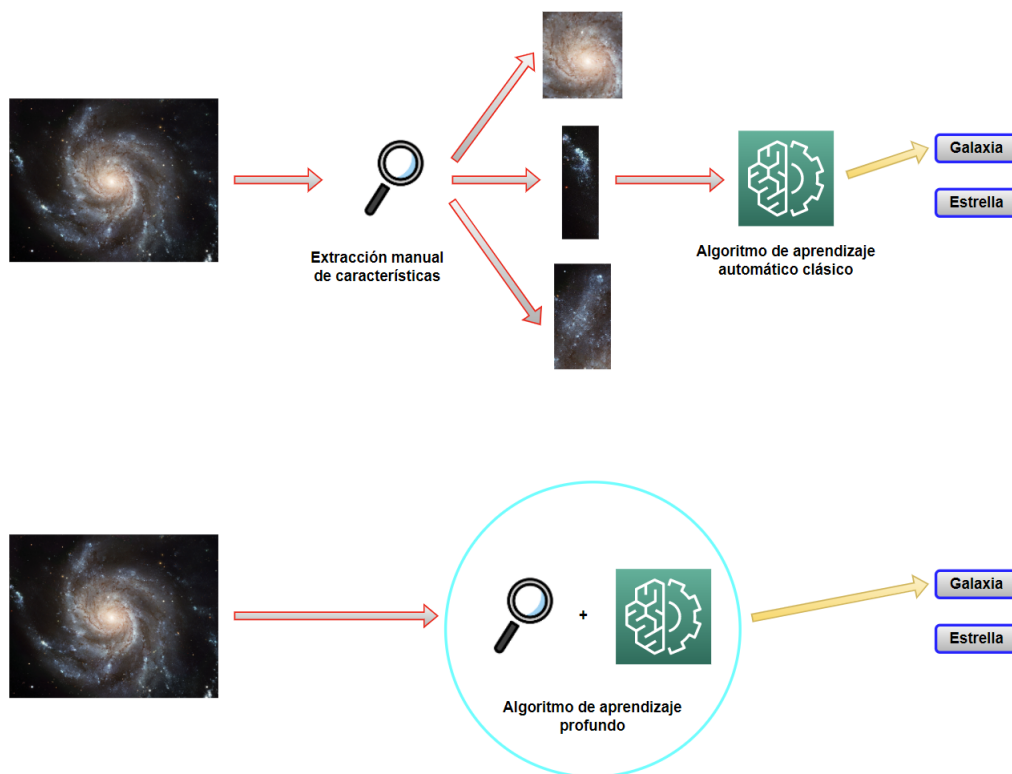


Figura 3.1: Extracción de características y aprendizaje implícitos en las aproximaciones *deep learning*.

Por el término aprendizaje profundo, o *deep learning*, puede entenderse como aquel subconjunto de las redes neuronales que cuentan con un gran número de parámetros ajustables, ya sea por disponer de multitud de capas o porque sus capas contengan un elevado número de unidades de procesamiento o neuronas. De esta manera, la complejidad inherente a los algoritmos de aprendizaje profundo permiten explotar la información oculta en conjuntos de datos masivos de una manera mucho más eficiente de lo que son capaces

las técnicas clásicas de aprendizaje automático con extracción manual de las características, además de automatizar y optimizar el trabajo de la selección manual de características (Figura 3.1).

Uno de los tipos de redes neuronales artificiales de aprendizaje profundo más utilizadas, y con aplicaciones más interesantes, son las redes neuronales convolucionales. Debido a que estos modelos serán los utilizados de manera intensiva este trabajo se presentarán con detalle en las secciones venideras del capítulo actual. Un diagrama de Venn mostrando la relación jerárquica entre los distintos conceptos que se han explicado en este comienzo del capítulo se encuentra en la Figura 3.2.

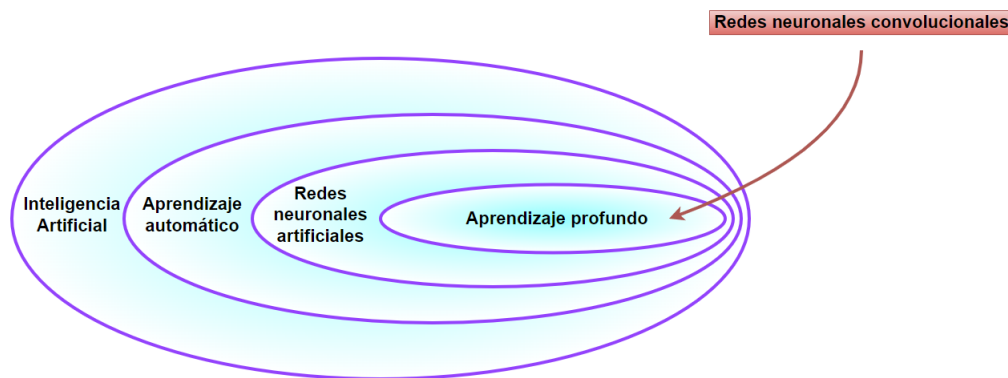


Figura 3.2: Diagrama de Venn mostrando la relación entre la inteligencia artificial y algunos de sus campos.

3.1. Conceptos generales

En esta sección se hablará acerca de la evolución y los conceptos más genéricos que componen las *Convolutional Neural Networks* (CNNs).

3.1.1 Evolución

Las redes neuronales convolucionales son un tipo de redes neuronales cuyo objetivo es replicar los procesos que ocurren en la corteza visual primaria de un cerebro biológico. Este enfoque confiere a este tipo de modelos la capacidad de obtener un rendimiento excelente en una amplia gama de tareas relacionadas con la visión artificial, como lo son la clasificación de imágenes, la segmentación de instancias, la detección de objetos o la generación de escenas. Tal es el rendimiento y las oportunidades que ofrecen las redes neuronales convolucionales que incluso han llegado a aplicarse a tareas de *speech recognition* o *Natural Language Processing* (NLP) [24].

Fukushima [8], [9], construyó una red neuronal que funcionaba de la misma manera que el cerebro humano. Esta se fundamentaba en 2 conceptos. Primero, las neuronas se replican a lo largo de toda la corteza visual. Segundo, existen células complejas que reúnen la información de células más simples. Como resultado, el desplazamiento de una imagen cambiaría la activación de las células primarias simples pero no influiría en la información sintetizada en las células complejas, concepto que daría pie a la introducción de las convoluciones y de las operaciones de submuestreo o *pooling*.

En años posteriores, LeCun [20] usó la técnica de retropropagación para entrenar una red neuronal convolucional para reconocer dígitos manuscritos. El reconocimiento de caracteres y patrones mediante un modelo entrenado de principio a fin era algo nuevo en aquella época, siendo utilizados anteriormente extractores de características mediante el uso de un modelo supervisado. Las nuevas redes convolucionales eran capaces de identificar diversos caracteres en una misma imagen. Para lograrlo, se utilizaba una pequeña ventana que se iba desplazando por toda la imagen, si en una región concreta la ventana se activaba significaba que en esa zona había presencia de un carácter.

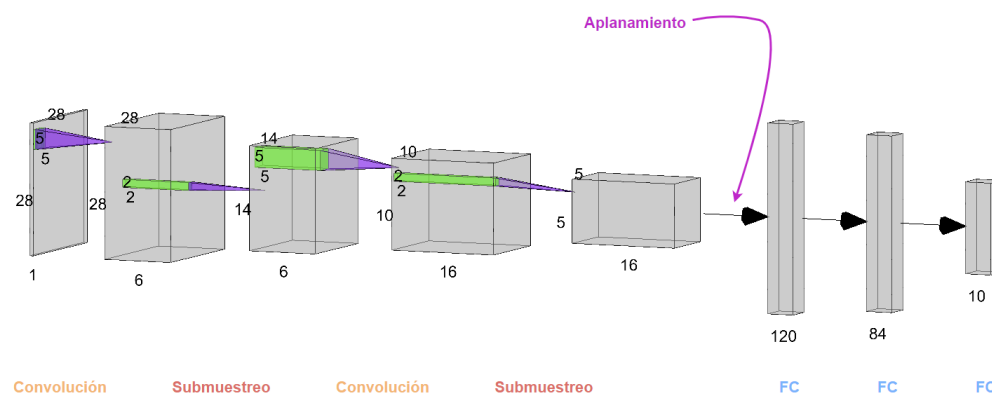


Figura 3.3: Esquema de la arquitectura de la red LeNet-5.

Esa misma idea fue utilizada muchos años después para resolver problemas de detección de caras o personas y para la segmentación semántica, esta es, la clasificación píxel a píxel. Estas aplicaciones se volvieron de gran utilidad en industrias como la conducción autónoma en la tarea de la identificación de carriles.

3.1.2 Arquitectura y operaciones

Para introducir y explicar los diferentes componentes y operaciones que componen una red neuronal convolucional prototípica se tomará como modelo base la red LeNet-5 (Figura 3.3), introducida por Yann LeCun en [21]. Esta red está compuesta por dos capas convolucionales, cada una de ellas formada por una operación de convolución y una operación de submuestreo o *pooling*, y 3 capas *Fully Connected* (FC).

3.1.2.1 Operación de convolución

La convolución tiene su origen y primera aplicación importante en la teoría de la señal, tratándose de un operador matemático que combina dos señales para formar una tercera. Sean dos funciones f y g , la convolución de ambas, $f * g$, se define como

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\eta) g(t - \eta) d\eta$$

Generalizando, una convolución se trata de un tipo de media móvil en la que se superponen f y una versión trasladada e invertida de g .

Trasladando el concepto a las redes neuronales convolucionales, el operador se implementa mediante la aplicación repetida de un filtro o *kernel* a un volumen de entrada, generando lo que se conoce como mapa de características o mapa de activaciones. Los hiperparámetros que caracterizan a las convoluciones son los siguientes:

- **Canales de entrada:** profundidad del volumen de entrada. Por ejemplo, una imagen en formato RGB de 24 bits tendrá 3 canales, tomando cada píxel en cada canal 2^8 posibles valores.
- **Canales de salida:** profundidad del volumen de salida. Una operación de convolución puede aprender diversos filtros, trasladándose en la generación de una profundidad variable del volumen de salida.
- **Tamaño de filtro:** tamaño del conjunto de pesos de la neurona convolucional que realizan la operación de convolución con el volumen de entrada de la neurona. Pueden ser simétricos o asimétricos.
- **Stride:** parámetro que rige el movimiento del *kernel* sobre el volumen de entrada. En el caso de que el *stride* valga $[1, 1]$, el *kernel* se deslizará sobre el volumen de entrada de una posición en una posición, de izquierda a derecha y de arriba a abajo. Si el *stride* vale $[1, 2]$, el *kernel* se deslizará sobre la imagen de píxel en píxel de izquierda a derecha, pero de 2 píxeles en 2 píxeles de arriba a abajo.
- **Dilatación:** técnica para expandir el filtro de tal manera que se insertan huecos entre los distintos parámetros del *kernel*.
- **Padding:** es la adición de posiciones nulas artificiales en los bordes de los volúmenes de entrada. Se utilizan para permitir que el *kernel* cubra una mayor parte del volumen y le resuma de una manera más adecuada al operar más veces sobre los píxeles que conforman los bordes del volumen original.

La manera numérica en la que los distintos elementos se interrelacionan configurando las dimensiones de los volúmenes de entrada y salida, en una convolución bidimensional, es la siguiente:

$$\mathcal{H}, \mathcal{W}_{\text{out}} = \frac{\mathcal{H}, \mathcal{W}_{\text{in}} + 2 \cdot \mathcal{P} - \mathcal{D} \cdot (\mathcal{K} - 1) - 1}{\mathcal{S}} + 1$$

donde $\mathcal{H}, \mathcal{W}_{\text{out}}$ es la altura y anchura del volumen de salida, $\mathcal{H}, \mathcal{W}_{\text{in}}$ es la altura y anchura del volumen de entrada, \mathcal{P} el *padding*, \mathcal{D} la dilatación, \mathcal{K} el tamaño del filtro y \mathcal{S} el *stride*.

En la Figura 3.4 se aprecia cómo la aplicación de distintos filtros 3×3 conocidos sobre un imagen de 8 bits de profundidad produce mapas de características diferentes, en función de los pesos que caractericen el *kernel*.

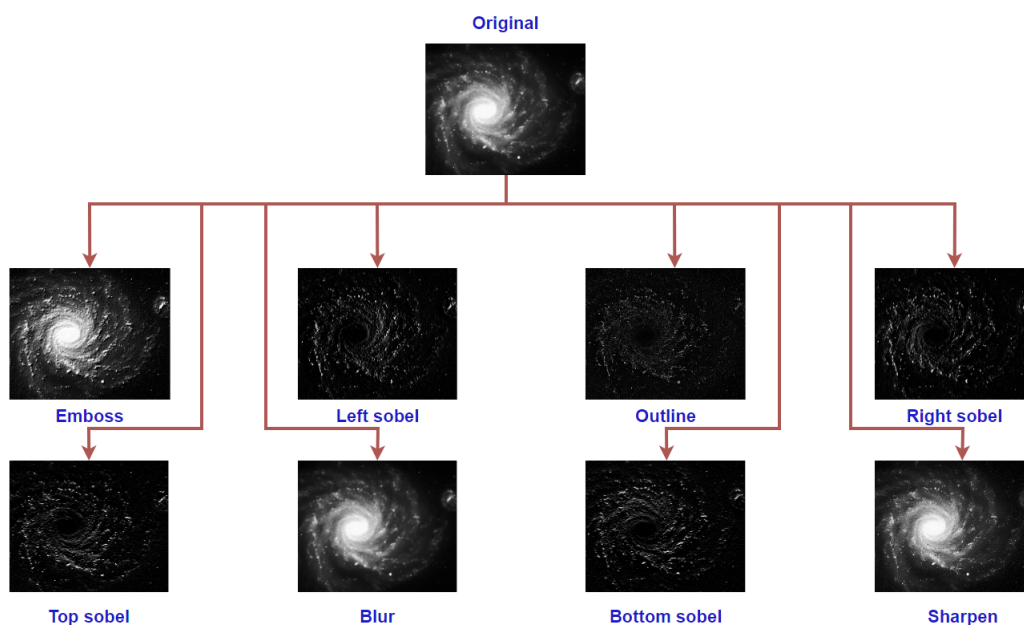


Figura 3.4: Efecto visual de la generación de mapas de características distintos en función del filtro que se aplique.

3.1.2.2 Operación de submuestreo

La operación de *pooling* o submuestreo permite reducir el tamaño de un mapa de características resumiendo las propiedades más importantes del mismo. Su aplicación es imprescindible por dos razones de peso:

- Se reducen los requisitos computacionales para ajustar los pesos de los modelos *deep learning* al reducir drásticamente el tamaño del volumen de los tensores.
- Por su naturaleza como operación de muestreo reduce o mitiga la aparición de sobreajuste que, por la manera en la que están construidas, ocurre frecuentemente durante

el entrenamiento de las redes neuronales convolucionales profundas. Las operaciones convolucionales recuerdan de manera exacta la posición de las características más importantes de las entradas, por lo que pequeñas variaciones de la posición de dichas características resultarán en cálculos diferentes (por ejemplo rotando una imagen o haciendo un ligero zum). Disminuyendo la resolución de la imagen mediante *pooling* se consiguen mantener las características más importantes de las entradas pero sin ser tan dependientes de su localización original.

Al igual que sucedía con la operación de convolución, el *pooling* también dispone de un filtro (conjunto de píxeles que se van a resumir), *stride* (rige el movimiento del filtro sobre el mapa de características) y *padding* (utilizado por si surge una incompatibilidad de dimensiones entre la combinación del filtro y el *stride* con las dimensiones del mapa de características). Las dimensiones del volumen de salida tras aplicar un operación de *pooling* a un volumen de entrada se pueden calcular con la misma fórmula que en el caso de la operación de convolución.

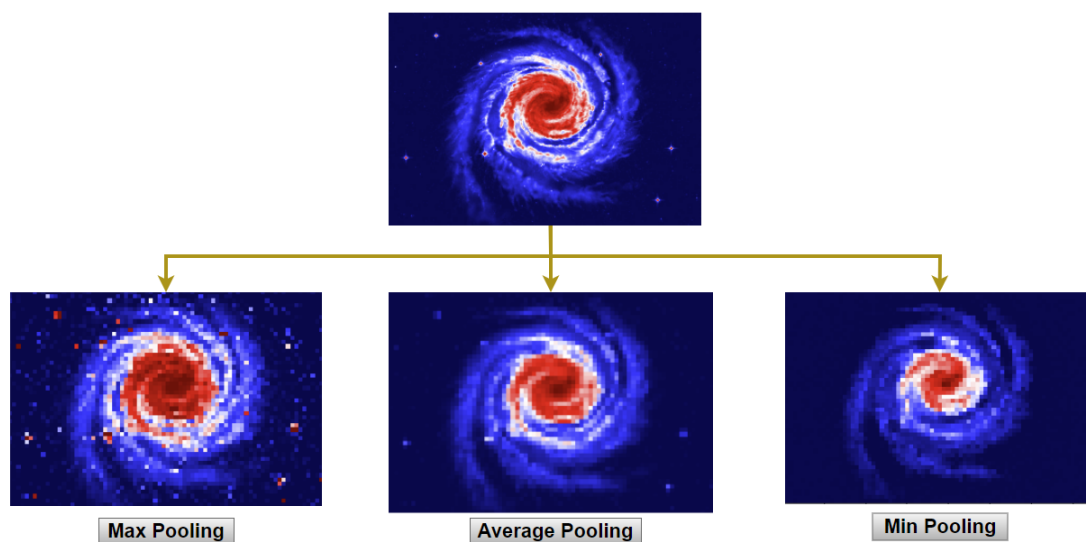


Figura 3.5: Efecto visual de la aplicación de *pooling* del máximo (izquierda), promedio (centro) y mínimo (derecha).

El submuestreo más comúnmente utilizado es el del máximo, seleccionando el mayor valor que cubra la ventana definida por el tamaño de *kernel*. No obstante, existen otros tipos como el promedio y el mínimo, generando su aplicación resultados diferentes tal y como se muestra en la Figura 3.5.

3.1.2.3 Capas densas

Una vez las capas convolucionales de la red han reducido el tamaño del mapa de características y extraído la información más relevante de los volúmenes de entrada

es necesario aplanar el volumen de salida de la última capa convolucional antes de su introducción a la primera capa FC. Si el volumen de salida está compuesto por α canales de dimensiones $\beta \times \gamma$, entonces de su aplanado se obtendrá como resultado un vector de $\alpha \times \beta \times \gamma$ posiciones.

Una vez se ha realizado el aplanado, las capas *fully connected* se corresponden con un perceptrón multicapa clásico. Todas las neuronas de dos capas consecutivas están conectadas entre sí, y en cada neurona se realiza el paso de la suma ponderada de sus entradas por una función de activación no lineal. La capa de entrada tendrá tantas neuronas como posiciones tenga el vector obtenido tras realizar el aplanado del volumen de salida de la última capa convolucional. La capa de salida tendrá tantas neuronas como se requieran para ajustarse al problema: en un problema de clasificación o regresión donde hay k categorías, la capa de salida tendrá k neuronas.

3.2. Equivarianza e invarianza

Uno de los conceptos más importantes de comprender acerca del funcionamiento de las redes neuronales convolucionales es su comportamiento frente a cambios en la traslación, la rotación y el escalado de las entradas de la red. La información explicada en esta sección ha sido adaptada de [11].

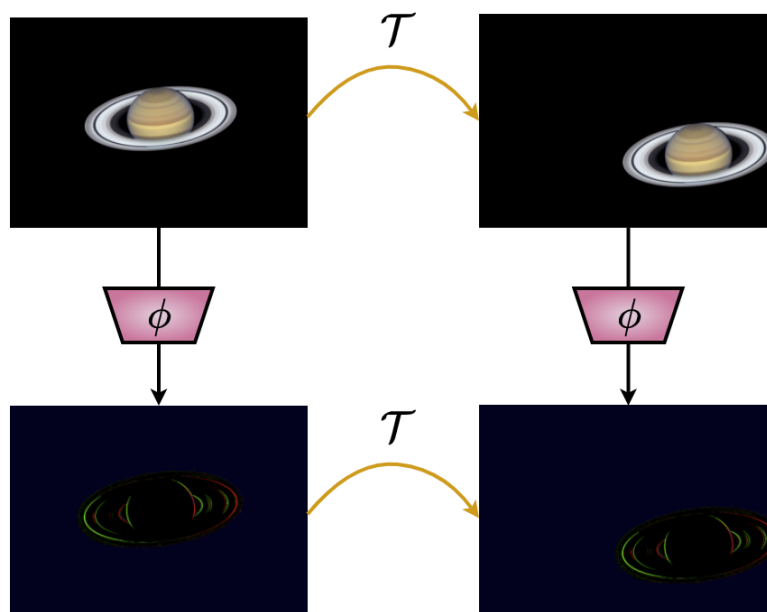


Figura 3.6: Equivarianza a la traslación de la operación de convolución.

Los conceptos de equivarianza e invarianza, a pesar de poder resultar similares, no hacen referencia a la misma propiedad. Para explicar las diferencias se va a recurrir al

concepto de la traslación. Un mapeo equivariante es un mapeo que preserva la estructura algebraica de una transformación. La operación de convolución es equivariante a la traslación, tal y como se puede apreciar en la Figura 3.6. Cuando una imagen sufre una traslación determinada por una matriz \mathcal{T} , entonces aplicando dicha matriz al mapa de características de la imagen original extraído por la convolución ϕ se obtiene el mapa de características, extraído por la misma convolución ϕ , de la imagen trasladada. La equivarianza de las convoluciones ha sido demostrada en el espacio de la frecuencia y a través de la transformada de Fourier, donde se encontró que las convoluciones en el dominio de las imágenes o en el dominio del tiempo son equivalentes a una multiplicación en el espacio de la frecuencia [18].

Por otro lado, la invarianza hace referencia a la insensibilidad frente a cierto suceso. Por ejemplo, una red neuronal invariante frente a la traslación es una red que no producirá una salida diferente únicamente porque el objeto localizado en el espacio de entrada haya sufrido una traslación lineal, tal y como se aprecia en la Figura 3.7.

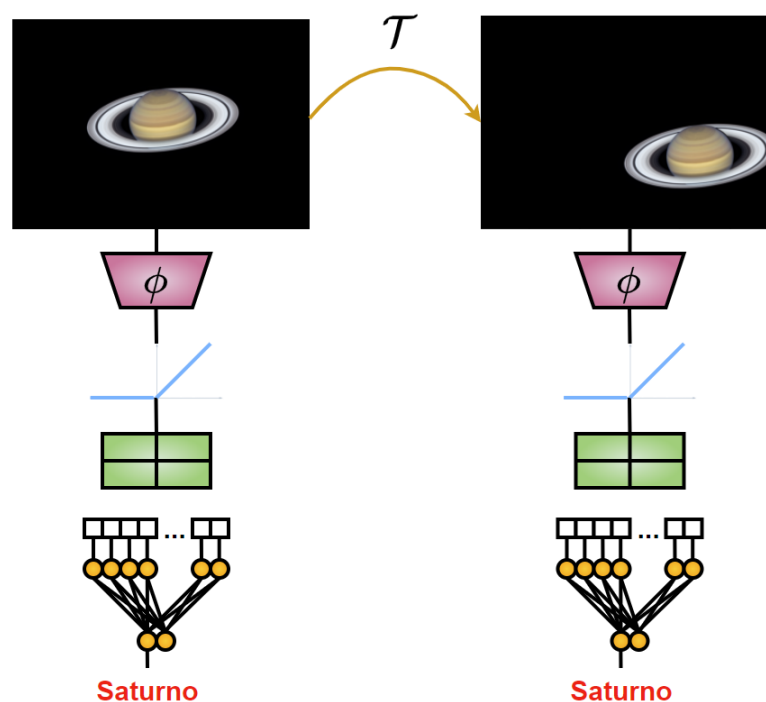


Figura 3.7: Invarianza a la traslación.

Las operaciones de convolución no son invariantes a la traslación dado que las salidas que producen ante imágenes trasladadas linealmente no son idénticas, mantienen la estructura algebraica pero el mapa de características obtenido no es el mismo. No obstante, las redes neuronales convolucionales si que son invariantes a la traslación. Esto es logrado gracias a la aplicación de operadores de submuestreo espacial. De manera local, las capas de *pooling* seguirán extrayendo los mismos máximos y mínimos ante pequeñas variaciones

de la traslación, los cuales siendo propagados hasta operadores más profundos de la red provocarán la invarianza frente a la traslación. Es necesario remarcar que algunos estudios recientes como [3] señalan la necesidad de adaptar ciertas operaciones y estructuras de las redes convolucionales para lograr una invarianza a la traslación real y resistente a desplazamientos muy marcados.

Una vez vistas las propiedades de las operaciones de convolución y las redes neuronales frente a la traslación se procederá a estudiar el comportamiento de las mismas frente a las rotaciones y los cambios de escala de las entradas. La convolución no es equivariante, de manera natural, con algunas otras transformaciones, como los cambios de escala o de rotación de una imagen. Se necesitan otros mecanismos para manejar este tipo de transformaciones. En la Figura 3.8 vemos una imagen de Saturno presentada a la red neuronal convolucional junto con un conjunto de filtros que la red ha aprendido en el proceso de entrenamiento. Hay un filtro dentro de la red que ha aprendido cómo es Saturno girado 70 grados, por lo que este se dispara y emite una fuerte activación. Dicha activación será posteriormente recogida durante la etapa de agrupación y submuestreo y, en última instancia, provocará que la red infiera que la imagen representa a Saturno.

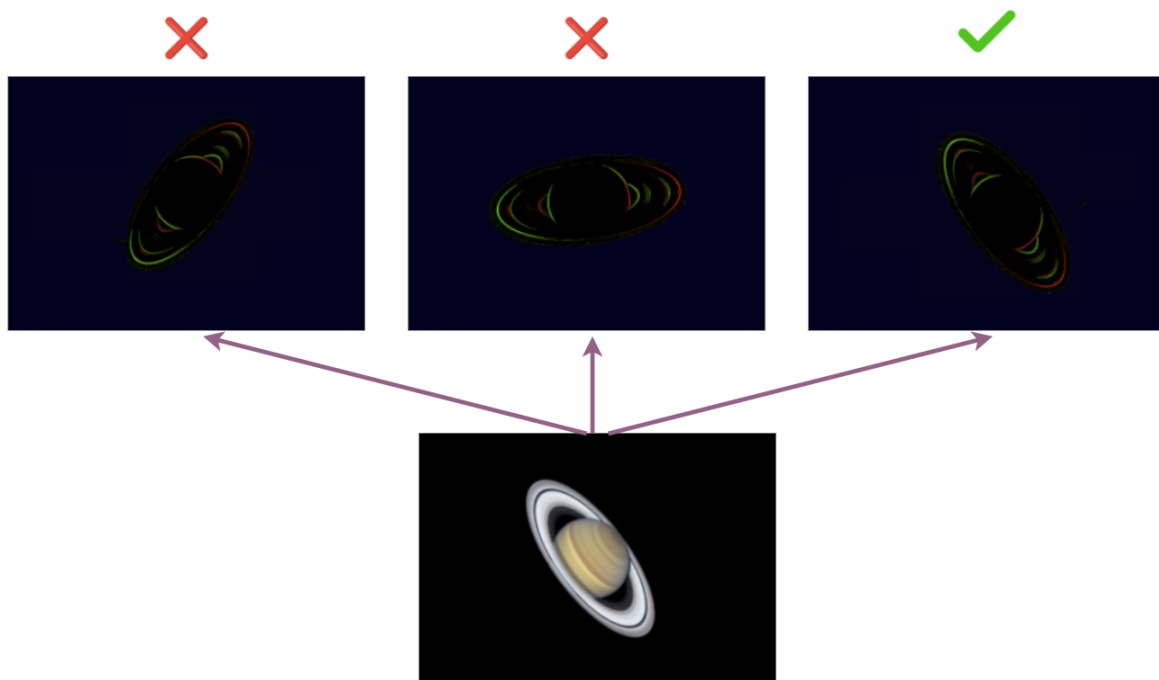


Figura 3.8: Invarianza a la rotación.

Para lograr que la red almacene filtros en los que aparezcan diferentes objetos de interés rotados es necesario que durante el proceso de entrenamiento de la misma se ofrezcan instancias donde se les ha aplicado dicha transformación. Esto se puede lograr de manera sencilla mediante generación de datos artificiales, en la técnica conocida como *data augmentation*. Respecto a los cambios de escala, el planteamiento es exactamente el mismo

que en el caso de las rotaciones. Mediante la introducción de instancias en diferentes escalas a la red durante el proceso de aprendizaje se desarrollarán diferentes filtros que podrán ser activados en la etapa de inferencia cuando aparezcan *inputs* en diferentes escalas.

3.3. Arquitecturas modernas

En esta sección se hablará acerca de varias arquitecturas de redes neuronales convolucionales que se encuentran en el estado del arte. En estadios más avanzados de este trabajo dichas arquitecturas serán adaptadas e integradas en modelos más complejos para detectar bordes de galaxias en el espacio distante. El texto seguido como referencia en esta sección se encuentra en [35].

3.3.1 ResNet

Las redes neuronales convolucionales residuales se introducen en [13]. La idea que persiguen los autores tras la creación de este tipo de redes es que la adición de nuevas capas a una arquitectura provoque que la correspondiente red sea más expresiva que su versión simplificada, esto es, con menos capas, en vez de simplemente lograr aprender un concepto diferente.

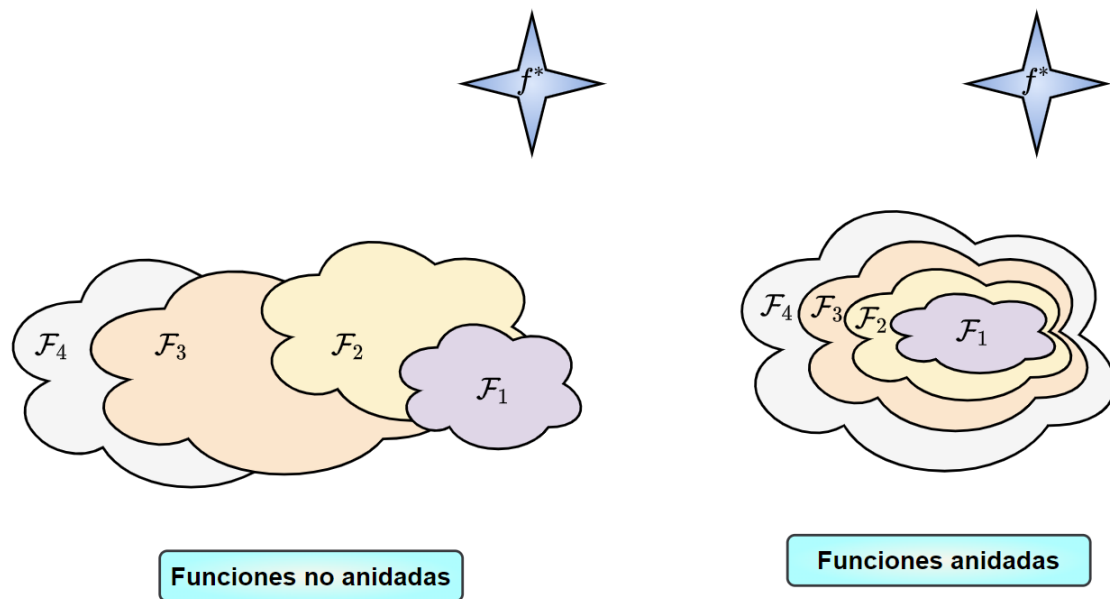


Figura 3.9: Funciones anidadas, no anidadas y función ideal.

Para explicar el concepto supongamos que \mathcal{F} representa el conjunto de funciones que una red neuronal específica puede alcanzar, teniendo en cuenta la relación de hiperparámetros que la definen. Esto significa que $\forall f \in \mathcal{F}$ existe un conjunto de parámetros, pesos y sesgos, que pueden ser obtenidos durante el entrenamiento a partir de algún conjunto de datos. Supongamos que f^* es la función verdadera que deseamos alcanzar mediante el

entrenamiento de la red. Si $f^* \in \mathcal{F}$ entonces la suerte está de nuestra parte, no obstante típicamente esto no será así. Por ello, trataremos de encontrar $f_{\mathcal{F}}^*$, la función más similar a f^* que esté en \mathcal{F} . Es razonable suponer que si diseñamos una arquitectura diferente y más potente \mathcal{F}' deberíamos llegar a un resultado mejor. En otras palabras, esperamos que $f_{\mathcal{F}'}^*$ sea mejor que $f_{\mathcal{F}}^*$. No obstante, no tenemos garantías que eso pueda ocurrir si $\mathcal{F} \not\subseteq \mathcal{F}'$.

Por ejemplo, en la Figura 3.9 se aprecia como para agrupaciones de funciones no anidadas puede darse el caso de que una agrupación de menor tamaño, \mathcal{F}_2 esté más cerca de la función f^* que agrupaciones de mayor tamaño como \mathcal{F}_3 o \mathcal{F}_4 . Con agrupaciones de funciones anidadas tenemos la garantía de que, al menos, \mathcal{F}_3 o \mathcal{F}_4 estarán igual de cerca de la función verdadera que \mathcal{F}_2 .

Este suceso fue tenido en cuenta durante el diseño de las redes neuronales convolucionales residuales, tratando de que cada capa adicional añadida a un modelo debe contener a la función identidad de una manera más sencilla que las capas anteriores. Esta consideración, a pesar de parecer compleja de alcanzar, indujo a una sencilla solución: los bloques residuales (Figura 3.10).

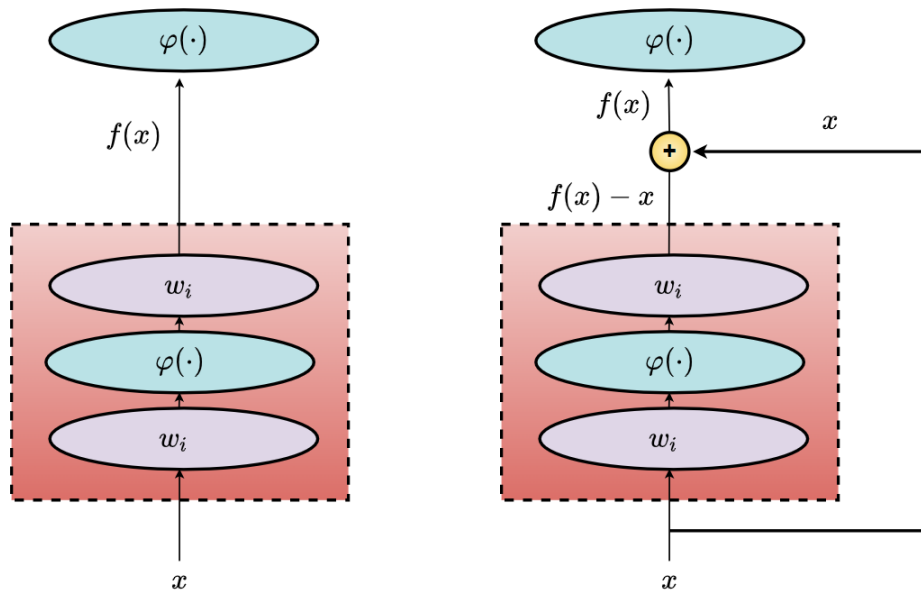


Figura 3.10: Bloque normal (izquierda) frente a bloque residual (derecha).

Supongamos x la entrada de los bloques. Sea el mapa subyacente deseado que queremos obtener mediante el aprendizaje del bloque $f(x)$, que se utilizará como entrada a la función de activación en la parte superior de la Figura 3.10. Un bloque regular debe aprender directamente el mapeo $f(x)$. No obstante, un bloque residual necesita aprender el mapeo residual $f(x) - x$. Si el mapeo de identidad $f(x) = x$ es el mapeo subyacente deseado, el

mapeo residual es más fácil de aprender que el mapeo regular: sólo tenemos que asignar a todos los pesos y sesgos del bloque el valor cero.

De esta manera, las redes neuronales convolucionales pertenecientes a la familia ResNet están conformadas por bloques residuales que, a su vez, siguen el diseño introducido por las redes *Visual Geometry Group* (VGG). El bloque residual consta de dos capas convolucionales de tamaño de filtro 3×3 con el mismo número de canales de salida. Cada capa convolucional va seguida de una capa de normalización de *batch* y una función de activación rectificador lineal. Tras el bloque, se añade al valor resultante del mismo la entrada original mediante la conocida como *skip connection*. Algunas de sus arquitecturas más conocidas son las de 18, 34, 50, 101, 152 y 200 capas ocultas, contando respectivamente con 11, 21, 25, 44, 60 y 64 millones de parámetros ajustables.

3.3.2 DenseNet

Las redes neuronales convolucionales densas se introducen en [17]. Para explicar su creación, los autores detectaron un problema que pueden llegar a tener las arquitecturas de la familia ResNet. Para explicar el problema es necesario revisar la descomposición de Taylor de las funciones.

$$f(x) = f(0) + f'(0)x + \frac{f''(0)x^2}{2!} + \frac{f'''(0)x^3}{3!}$$

Una función se descompone en términos de orden creciente. De forma similar, ResNet descompone las funciones en un término lineal sencillo y un término no lineal más complejo:

$$f(x) = x + g(x)$$

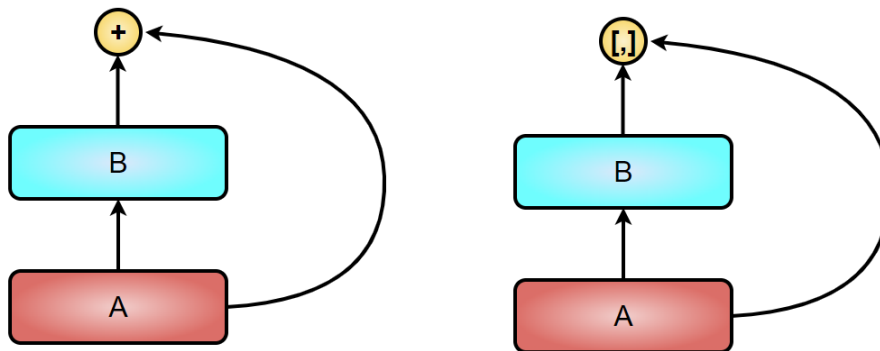


Figura 3.11: Bloque ResNet (izquierda) frente a bloque DenseNet (derecha).

Si se desea resumir y combinar la información de ambos términos, en vez de simplemente sumarla como se hace en los bloques ResNet, se llega a la solución que propone la familia

de redes DenseNet: la concatenación (Figura 3.11). La principal diferencia entre ResNet y DenseNet es que en este último caso las salidas se concatenan en lugar de sumarse. De esta manera, se logra un mapeo resultante formado por una compleja combinación de la entrada inicial:

$$x \rightarrow [x, f_1(x), f_2([x, f_1(x)]), f_3([x, f_1(x), f_2([x, f_1(x))])], \dots]$$

Finalmente, todos esos términos son combinados en un perceptrón multicapa para reducir el número de características. Algunas de las arquitecturas más conocidas bajo la familia DenseNet son DenseNet-121, DenseNet-161, DenseNet-169 y DenseNet-201, siendo un número mayor indicativo de una mayor profundidad de la red. El número de parámetros entrenables de cada configuración es de 7 millones, 28 millones, 14 millones y 20 millones.

3.3.3. EfficientNet

Las redes neuronales convolucionales de la familia EfficientNet se introducen en [32] con el objetivo de presentar un nuevo marco para realizar un escalado inteligente de los modelos.

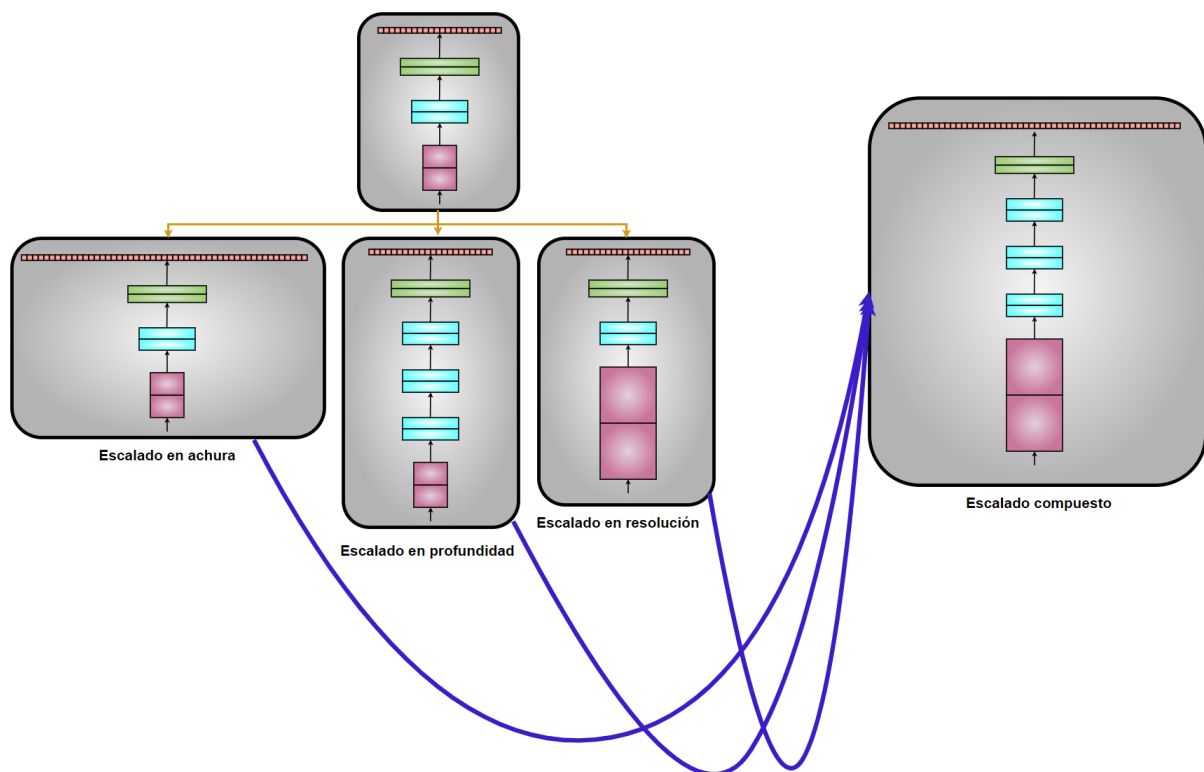


Figura 3.12: Escalados en anchura, profundidad, resolución y compuesto.

Típicamente, los modelos de redes neuronales profundos crecen de tres maneras posibles:

- Aumenta el número de canales, esto es, un escalado en anchura.
- Aumenta el número de operaciones y capas, esto es, un escalado en profundidad.
- Aumenta la resolución de los tensores, esto es, un escalado en resolución.

Las redes de la familia EfficientNet han sido concebidas para realizar un escalado compuesto, esto es, manteniendo una relación proporcional entre el escalado en anchura, en profundidad y en resolución, conociéndose el resultado como escalado compuesto, tal y como se aprecia en la Figura 3.12. Los autores consiguieron mediante esta técnica redes neuronales con mas precisión y menos hiperparámetros que el resto de arquitecturas que se encontraban en el estado del arte en el momento de la publicación.

Algunas de las arquitecturas más conocidas de la familia EfficientNet son la B0, con 5 millones parámetros, la B1, con 7 millones parámetros, la B2, con 9 millones parámetros, la B3, con 12 millones parámetros, la B4, con 19 millones parámetros, la B5, con 30 millones de parámetros, la B6, con 43 millones parámetros, la B7, con 66 millones de parámetros y la B8, con 87 millones de parámetros.

3.3.4. DPN

Las redes neuronales convolucionales de la familia *Dual Path Networks* (DPN) se introducen en [7] con el objetivo de integrar la reutilización de características que es lograda por la topología de conexiones incluidas en la familia ResNet y el descubrimiento de nuevas características promovido por la topología de conexiones que forman las redes de la familia DenseNet. Las redes de rutas duales han ganado competiciones como ILSVRC, y con ligeras modificaciones también alcanzan resultados en el estado del arte para las tareas de detección de objetos y segmentación semántica.

Para esquematizar de una manera sencilla el funcionamiento de las redes de ruta dual se representan las redes neuronales de la familia ResNet, DenseNet y DPN como *High Order Recurrent Neural Networks* (HORNN) (Figura 3.13). Dicha representación trata de mostrar la evolución histórica y la reminiscencia de conceptos provenientes de capas anteriores en cada capa de la red. De esta manera, se aprecia fácilmente la reutilización de características que permite la topología de conexiones internas que conforman las redes ResNet. También se observa la facilidad con la que las redes de la familia DenseNet generan nuevos conocimientos auxiliares a los que ya se han ido aprendiendo en estadios anteriores de la red, concatenándolos e incrementando la variedad y complejidad de representaciones generadas. El objetivo de diseño de las redes de ruta dual es el de mantener la capacidad que tienen las redes residuales para reutilizar antiguos conocimientos en conjunción con la flexibilidad que otorga la topología de conexiones internas de DenseNet para aprender representaciones muy variadas.

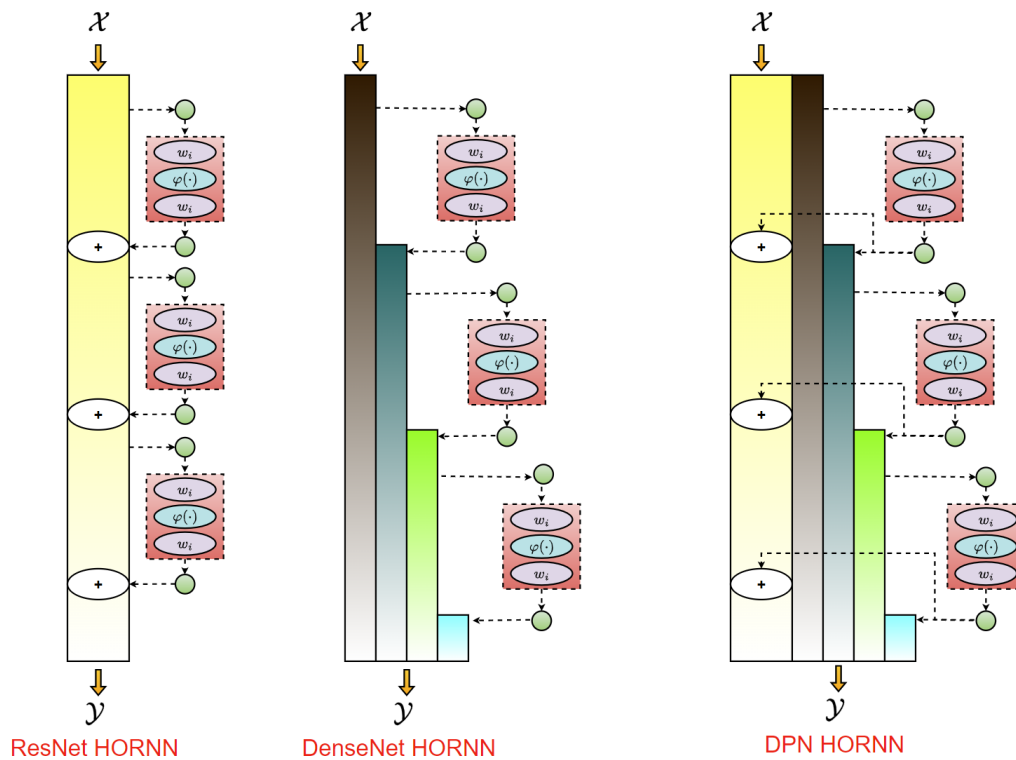


Figura 3.13: Representaciones de arquitecturas ResNet, DenseNet y de ruta dual como redes neuronales recurrentes de alto nivel.

Algunas de las arquitecturas más utilizadas de la familia de redes de ruta dual son DPN68, DPN92, DPN98, DPN107 y DPN131. Cada configuración cuenta con 12, 37, 61, 86 y 79 millones de parámetros ajustables respectivamente.

3.4. Segmentación semántica

Una vez se han presentado arquitecturas en el estado del arte de redes neuronales convolucionales profundas es el momento de hablar de un tipo de redes que, formando parte de su cuerpo, integran redes neuronales convolucionales pero cuyo cometido va más allá de atacar un problema de clasificación o regresión a partir de imágenes, típico caso de uso de las **CNNs**. Dichos modelos, que pueden pertenecer a varias familias diferentes, fueron concebidos para afrontar la tarea de la segmentación semántica, destacando entre ellas la familia de las redes en U o *U-Nets*, la familia de las redes en U anidadas o *U-Nets ++* y la familia de redes DeepLabV3, están orientados a resolver el problema de la segmentación semántica.

La segmentación semántica se enfoca en clasificar cada uno de los píxeles que haya en las imágenes, tratando el caso bidimensional, siendo el objetivo asignar una misma

categoría a aquellos píxeles que formen parte de un mismo objeto. Existen otros problemas comunes en visión artificial como lo son el reconocimiento de objetos, la detección o la segmentación de instancias, esto es, detección mas segmentación (Figura 3.14).

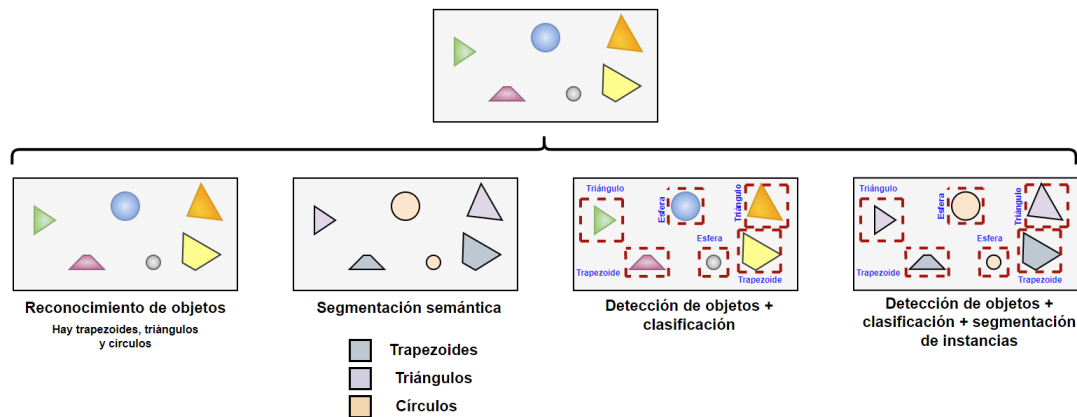


Figura 3.14: Algunas de las categorías de problemas más comunes en visión artificial.

En este trabajo, la segmentación semántica será el problema entorno al que pivotará toda la experimentación realizada, concretamente con el objetivo de detectar bordes y límites de galaxias en el espacio distante. Más detalles de cómo será afrontado el problema se ofrecerán en estadios más avanzados del trabajo. A continuación, se irán explicando aquellas arquitecturas más comúnmente utilizadas para abordar el problema a tratar.

3.4.1 Redes en U

Las *U-Nets*, o redes en U, fueron introducidas en [27] con el objetivo de atacar el problema de la segmentación semántica en el contexto de las imágenes biomédicas. Están compuestas por una etapa de contracción, llevada a cabo por un *encoder* de estructura similar al de una red neuronal convolucional clásica, y una etapa de expansión simétrica a la de contracción, llevada a cabo por un *decoder*. Las capas de contracción y expansión que se encuentran al mismo nivel presentan una comunicación horizontal denominada *skip connection*, doblando de esta manera el número de canales de cada capa del *decoder*. Este tipo de redes, tal y como argumentan sus autores, son capaces de obtener unas segmentaciones de calidad razonable con un tamaño muestral de entrenamiento relativamente pequeño.

Los *encoders*, esto es, la primera parte de una *U-Net*, realizan una fase contracción en la cual, típicamente mediante una serie de operaciones de convolución y submuestreo apiladas, se reduce el tamaño de los *inputs* de la red. De esta manera, el objetivo de esta primera parte de la red es reducir la resolución de las entradas. Por su parte, los *decoders* tienen como objetivo aumentar la resolución de las imágenes. Para lograrlo, típicamente hacen uso de las operaciones de convolución transpuestas y de *upsampling*. Conectando

capas simétricas en el *encoder* y en el *decoder* se encuentran las conexiones transversales denominadas *skip connections*. Cada salida de un bloque de contracción es añadida a la entrada de su correspondiente bloque de expansión, exceptuando la salida del primer bloque de expansión. Una representación gráfica de una U-Net se encuentra en la Figura 3.15.

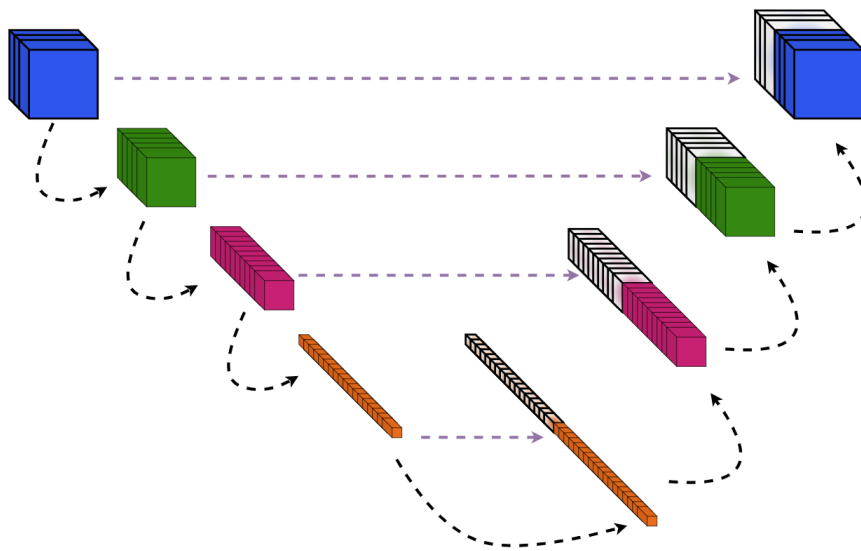


Figura 3.15: Arquitectura de una red en U.

3.4.2 Redes en U anidadas

Las redes en U anidadas, *nested U-Nets* o *U-Nets++* fueron introducidas en [36] como una potente modificación de las redes en U con el objetivo de realizar segmentación semántica en el ámbito de las imágenes biomédicas. De igual manera, este tipo de redes están compuestas por una arquitectura basada en un *encoder* y un *decoder*. La principal diferencia con respecto a sus antecesoras se encuentra en las conexiones transversales. Estas sufren un rediseño mediante tanto la inclusión de operaciones de convolución intermedias como un aumento en el número de conexiones entre capas (Figura 3.16). Por ejemplo, en la Figura 3.17 se aprecia como la conexión entre las capas $X^{0,0}$ y $X^{0,4}$ puede verse como un bloque denso, como los explicados en la sección 3.3.2, compuesto por 3 operaciones de convolución en los que además se tiene en cuenta información proveniente de capas más profundas de la red.

Debido al mayor número de operaciones de convolución y al flujo de una cantidad de información mayor entre capas las redes de la familia *U-Net++* presentan un número de parámetros bastante superior al de las redes en U clásicas, aún usando los mismos esquemas para codificar y expandir la información.

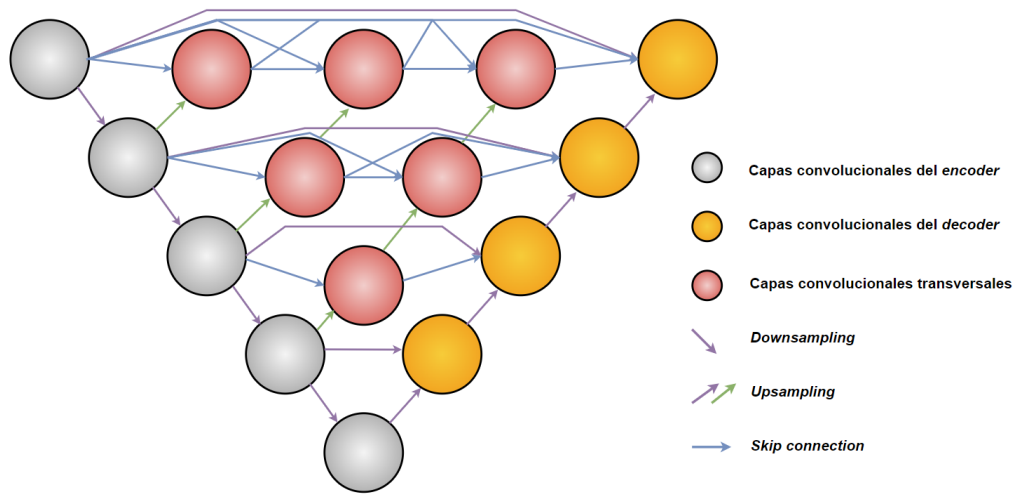


Figura 3.16: Arquitectura de una U-Net++.

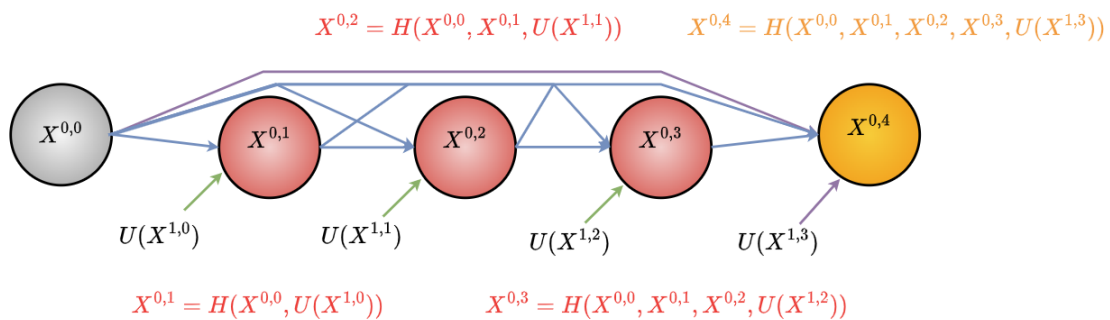


Figura 3.17: Conexión transversal rediseñada en una U-Net++

3.4.3 Arquitecturas con convoluciones dilatadas separables: DeepLabV3

DeepLabV3 es una familia de redes neuronales de segmentación semántica presentada por investigadores de Google en [6] con el objetivo de, mediante el uso de conceptos en el estado del arte aplicados a las redes neuronales convolucionales, superar a los modelos más precisos hasta en el momento en los *benchmarks* más conocidos donde se evalúan los modelos de segmentación. Los autores se dieron cuenta de que uno de los inconvenientes que pueden surgir mediante la aproximación a la segmentación semántica utilizando redes neuronales convolucionales profundas como *encoders* es que, a medida que el mapa de características se va reduciendo conforme los datos van fluyendo por el *encoder*, la información relativa a los objetos pequeños puede desaparecer, induciendo máscaras de segmentación en las que no aparecen las superficies ocupadas por dichos cuerpos.

Para mitigar ese problema, se recurre al concepto de las *atrous convolutions* o convoluciones dilatadas, una operación que permite tanto ajustar el campo de visión de cada operación de convolución como controlar la resolución de los mapas de características computados por cada capa. Las convoluciones dilatadas introducen un parámetro denominado radio de dilatación, el cual define un espaciamiento entre los distintos valores que componen el filtro o *kernel* de la convolución. Por ejemplo, un filtro cuadrado de *kernel* 3 en una convolución clásica tendrá un campo de visión de 3x3 píxeles. No obstante, si introducimos una dilatación de radio 2, el campo de visión aumentará al de una convolución clásica de *kernel* 5x5 (Figura 3.18), contando con un número menor de parámetros. Si la dilatación es de radio 3, entonces el campo de visión será el equivalente al de una convolución clásica de *kernel* 7x7. Un radio de 1 provoca la equivalencia en el comportamiento entre las convoluciones clásicas y las convoluciones dilatadas.

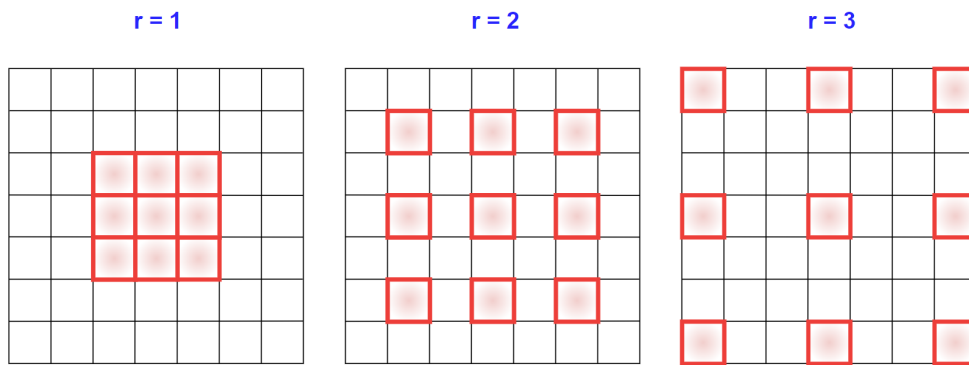


Figura 3.18: Convoluciones dilatadas con diferentes radios y *kernel* 3x3.

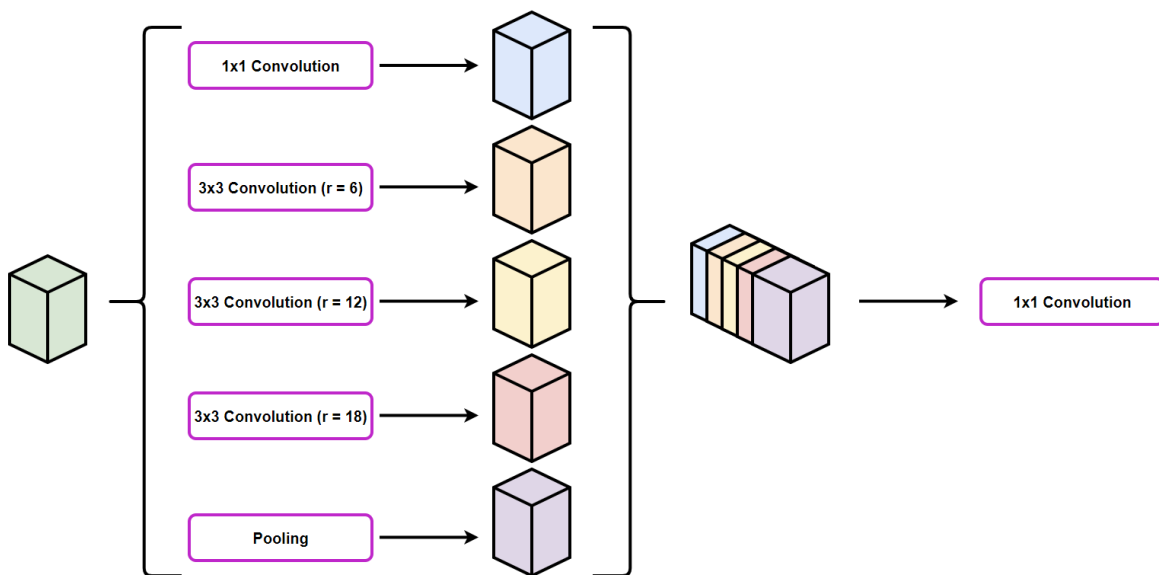


Figura 3.19: *Atrous Spatial Pyramid Pooling*.

De esta manera, las convoluciones dilatadas son utilizadas en DeepLabV3 debido a que los rasgos de objetos de muy diferente escala o tamaño son mejor preservados en los mapas de características producidos por dichas operaciones, tal y como se estudia en [16] y [10]. Los autores diseñan módulos que emplean convoluciones dilatadas en cascada y en paralelo para capturar el contexto de la información a múltiples escalas mediante el uso de diferentes radios de dilatación.

Los módulos utilizados para combinar las convoluciones dilatadas en paralelo son denominados *Atrous Spatial Pyramid Pooling (ASSPP)*, los cuales fueron originalmente propuestos en [5]. Dicho módulo trata de extraer diferentes campos de visión de manera paralela de un mismo mapa de características, combinándolos posteriormente con el objetivo de capturar la esencia de objetos en diferentes escalas (Figura 3.19).

La combinación de todos los conceptos anteriores en una arquitectura *encoder - decoder* otorga flexibilidad para utilizar distintos esquemas de redes neuronales convolucionales profundas como *encoders*, siempre que puedan adaptarse para soportar el uso de convoluciones dilatadas. De este modo, se obtiene la arquitectura DeepLabV3, representada en la Figura 3.20.

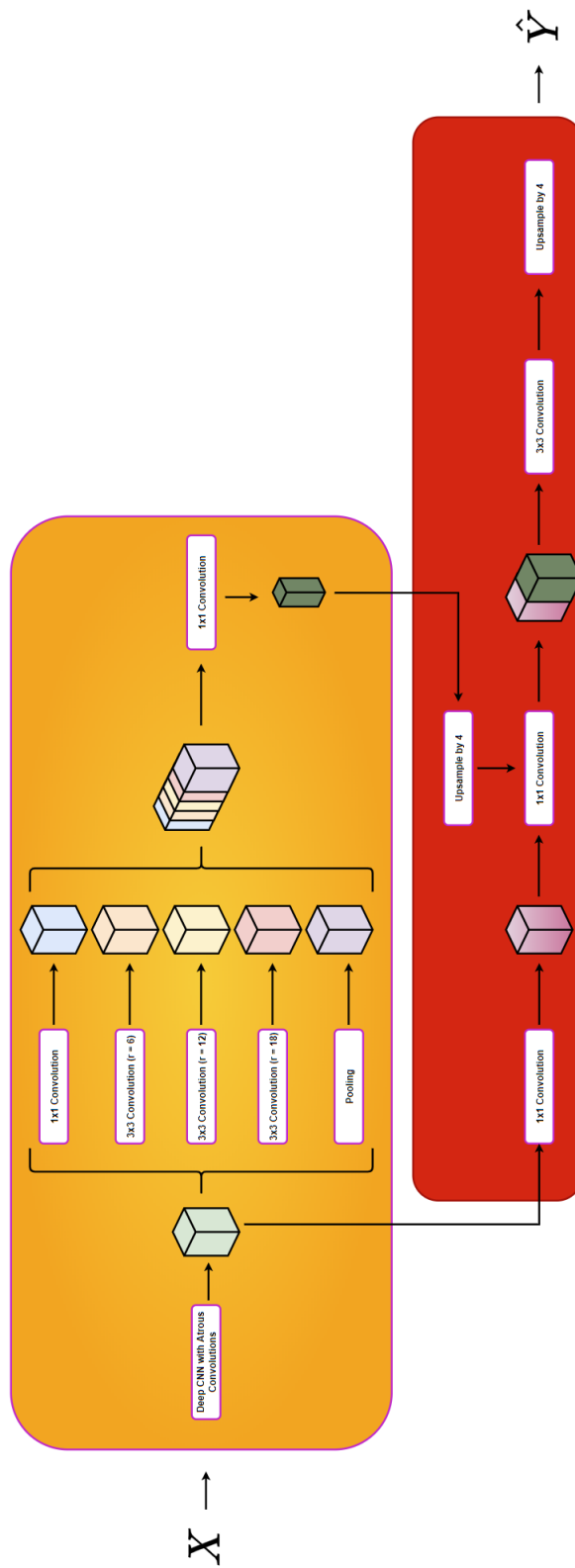


Figura 3.20: Arquitectura DeepLabV3.

4: Truncaciones astronómicas

Uno de los problemas que más ha perdurado en la comunidad astronómica y astrofísica es el de asignar un tamaño a una galaxia. El problema es complejo, en parte porque las galaxias son cuerpos difusos sin bordes nítidos ni caídas evidentes de la densidad de los objetos que las componen. Además, cuanto mayor es el tiempo de exposición utilizado para capturar información de una galaxia se percibirán partes más débiles de la misma, pareciendo el cuerpo más grande.

En la Figura 4.1 se aprecian 4 tomas de la misma galaxia con diferentes tiempos de exposición. La primera toma (parte superior izquierda) ha sido observada con el tiempo normal de exposición usado en el cartografiado de galaxias más utilizado en el universo

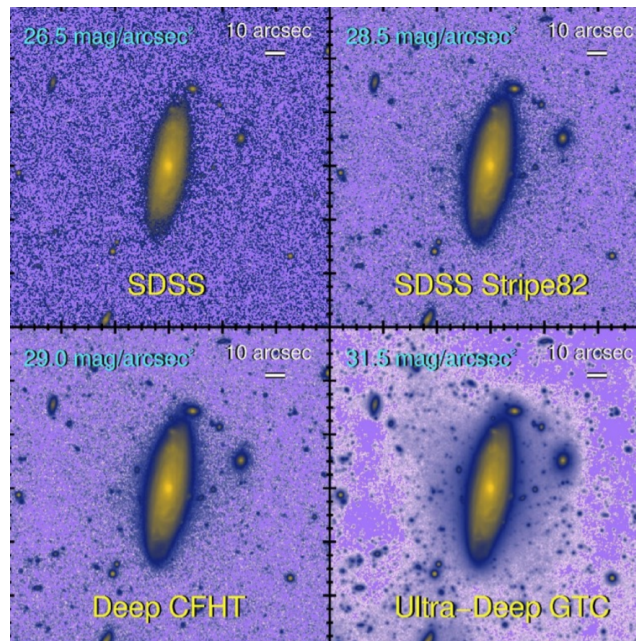


Figura 4.1: Tamaño de las galaxias en función del tiempo de exposición.

local. Se aprecia una galaxia de tipo espiral con unos pocos objetos (galaxias satélites) a su alrededor. Sin embargo, la misma galaxia si es observada durante un mayor tiempo de exposición y con el Gran Telescopio de Canarias (GTC), el telescopio óptico-infrarrojo más grande del mundo, se observan muchos más objetos vecinos a la galaxia (parte inferior derecha). Además, la galaxia pasa de ser un disco perfecto a tener cierta estructura en sus límites.

A continuación se ofrecerán tanto unas pinceladas acerca de los enfoques tradicionalmente aceptados para delimitar los bordes de las galaxias como una explicación detallada de la aproximación utilizada para calcular los bordes utilizados en este trabajo.

4.1. Enfoques clásicos

Durante el S.XX se han tratado de ofrecer diversas aproximaciones al cálculo de los límites de las galaxias [4], siendo los radios efectivos y los diámetros isofotales aceptados tradicionalmente como medidas del tamaño de las mismas. La primera discusión del tamaño aparente de las galaxias fue introducida por E. Hubble en el año 1926, el cual utilizó la ubicación de las isofotas de bajo brillo superficial para definir el diámetro de las galaxias. Las isofotas pueden definirse como contornos de brillo superficial constante, es decir, una curva sobre una superficie iluminada que relaciona puntos de igual brillo. Un ejemplo del cálculo de isofotas sobre una galaxia simulada realizado a partir de las bibliotecas *software* `astropy` y `photutils` de Python puede encontrarse en la Figura 4.2.

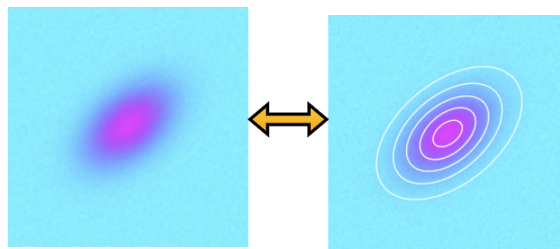


Figura 4.2: Cálculo de isofotas en una galaxia simulada.

Redman, en el año 1936, propuso el uso de una isofota de brillo superficial fija a 25 mag/arcsec^2 como definición del diámetro para galaxias elípticas. La magnitud es una medida del brillo superficial de objetos, por ejemplo, una estrella de magnitud 6 es más brillante que una estrella de magnitud 11. Por su parte, el término arcosegundo proviene de la división de un arco en segundos. Un círculo se divide en 360 grados, cada grado se divide en 60 minutos, y cada minuto se divide en 60 segundos. Por tanto, un arcosegundo al cuadrado tiene un área angular de un segundo por un segundo. De esta manera, la medida mag/arcsec^2 determina el brillo superficial aparente, o flujo, presente en una región del cielo.

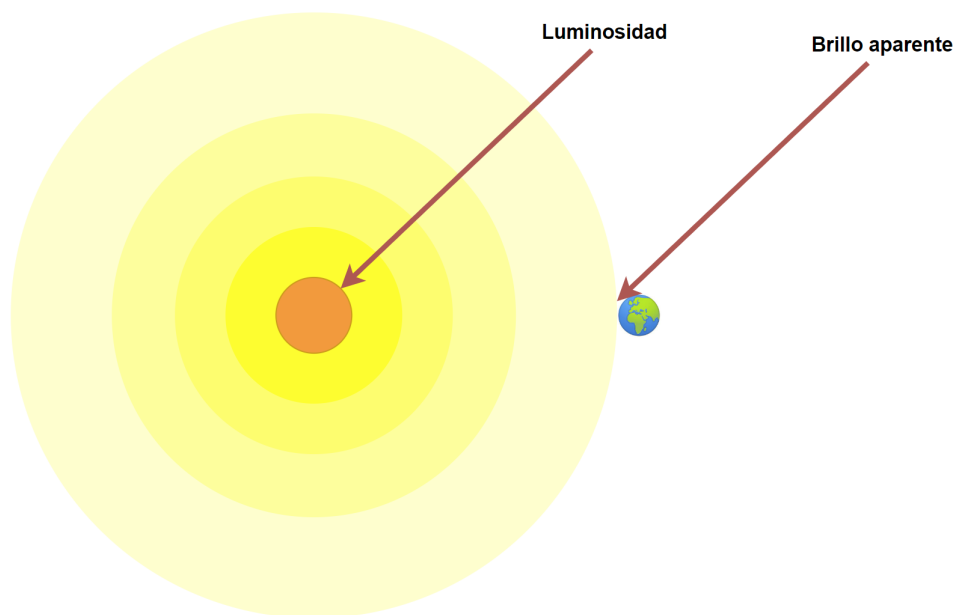


Figura 4.3: Luminosidad y brillo aparente de un cuerpo.

Dos conceptos importantes de distinguir son el de brillo, o magnitud, aparente y el de luminosidad. Mientras que la luminosidad hace referencia a la cantidad de energía que es irradiada desde la superficie de un cuerpo a través de todas las longitudes de onda el brillo aparente se refiere a la cantidad de energía que es detectada por un observador en la Tierra, tal y como se aprecia en la Figura 4.3. La luminosidad depende de algunos factores como la temperatura del cuerpo en cuestión, y no puede ser determinada por observación directa, mientras que la magnitud aparente puede ser calculado por observación directa. Para relacionar ambos conceptos se parte de la siguiente fórmula [25]:

$$\mathcal{F} = \frac{\mathcal{L}}{4\pi d^2}$$

donde F es el flujo que recibimos del objeto en un determinado rango del espectro electromagnético, \mathcal{L} la luminosidad y d la distancia entre el foco de emisión y el foco de recepción de energía. Una vez sabida la relación, la magnitud aparente es calculada como:

$$\text{mag}_{\text{aparente}} = -2,5 \log_{10} (\mathcal{F}) + \text{zeropoint.}$$

y la magnitud superficial como:

$$\text{mag}_{\text{superficial}} = -2,5 \log_{10} (\bar{\mathcal{F}}) + \text{zeropoint} + 5 \log_{10} (p_i).$$

donde $\bar{\mathcal{F}}$ es el flujo medio y p_i es la escala de los píxeles (utilizada para calcular el área del objeto en cuestión). Mientras que la magnitud aparente calcula cuánta luz en total

proviene de un objeto en un determinado rango electromagnético el brillo, o magnitud, superficial nos da cuánta luz nos llega en un determinado rango electromagnético por cada unidad de área de dicho objeto.

Esta aproximación de fijar un valor constante de isofota tiene algunos inconvenientes. Por ejemplo, puede verse influido por la presencia de múltiples galaxias cercanas entre sí, afectando la energía irradiada por una a los límites definidos para su vecina. Otra desventaja es la necesidad de fijar un criterio arbitrario, un valor estático, con su consiguiente rigidez y falta de adecuación a escenarios no contemplados para la determinación de ese valor.

Más recientemente, el semieje mayor de la elipse, también conocido como *half-light radius* o r_e , que abarca la mitad de la luz de una galaxia ha sido el indicador más utilizado (Figura 4.4). Las principales razones que han apoyado esta decisión son dos:

- Su robustez frente a la variación de parámetros y sucesos como la relación señal-ruido y el tiempo de exposición.
- Su conexión con el ajuste paramétrico del brillo superficial de las galaxias

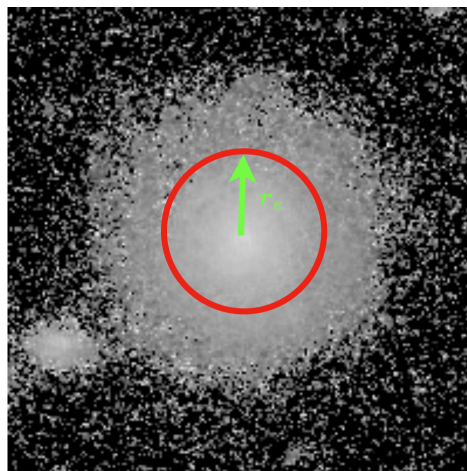


Figura 4.4: *Half-light radius*.

No obstante, esta aproximación no carece de problemas y sufre unos hándicaps muy similares a los de las medidas más tradicionales. Por un lado, se establece un criterio arbitrario al fijar en un medio la cantidad de luz que ha de ser recogida y, por otro lado, el cálculo está profundamente relacionado con la concentración de galaxias en la zona de interés.

4.2. Enfoques modernos

Ambas técnicas comentadas en la sección anterior se basan en la elección de criterios arbitrarios y fueron introducidas cuando las imágenes de galaxias que se disponían eran mucho menos profundas que las actuales. Por ese motivo, en [33] se ofrece una definición del tamaño de las galaxias con una motivación física basada en la localización del umbral de densidad del gas necesario para la formación de estrellas en las galaxias. Esta aproximación define un concepto de tamaño que está relacionado con los límites o bordes de los objetos en los que se produce un fuerte contraste en sus propiedades inherentes. Por esta razón, se utiliza como medida de tamaño la región donde teóricamente se espera que descienda la formación estelar dentro de una galaxia, esto es, que ocurra una truncación en sus perfiles de luz.

Para cuantificar y materializar el concepto, se elige el umbral de densidad de masa estelar de 1 Msun/pc^2 , $R1$, porque dicho valor es la densidad en el lugar del truncamiento de las galaxias similares a la Vía Láctea, donde Msun/pc^2 hace referencia a la masa solar por *parsec* cuadrado, siendo *parsec* cuadrado una medida de área formada por dos lados de 31 billones de kilómetros. De esta manera, $R1$ es un valor umbral de densidad de masa obtenido mediante la conversión de los perfiles de color (luz en múltiples longitudes de onda) de las galaxias en perfiles de relación masa-luz, es decir, convirtiendo la luz en masa de forma radial. Estos perfiles dan lugar a un perfil de densidad de masa estelar y se utilizan para determinar la ubicación de $R1$. La Figura 4.5 compara la ubicación de la medida de tamaño $R1$ y el radio efectivo r_e para una serie de galaxias. Es bastante evidente que $R1$ resume de manera más razonable lo que el nuestro sistema de visión identifica como el límite de una galaxia.

Más recientemente, en [2], se realiza un estudio con el objetivo de detectar truncamientos de galaxias, desde el universo local hasta grandes distancias cósmicas, superando el progresivo oscurecimiento cosmológico en el desplazamiento al rojo y con la restricción de necesitar una excelente resolución espacial. Por ello, se utiliza el *Low Surface Brightness (LSB)* en las imágenes profundas del *Hubble Space Telescope (HST)*. A partir de los perfiles de brillo, color y masa se pueden detectar los truncamientos de galaxias, R_{edge} , y también derivar la densidad estelar en esas posiciones, manteniendo una conexión con la aproximación seguida en [33].

De igual manera, se observa como la truncación situada en R_{edge} resulta mucho más adecuada, a efectos del ojo humano, que el radio efectivo r_e (Figura 4.6). Dichas truncaciones serán las que tratarán de aprender, y generalizar a nuevas galaxias, los algoritmos de aprendizaje profundo que se desarrollan en este trabajo.

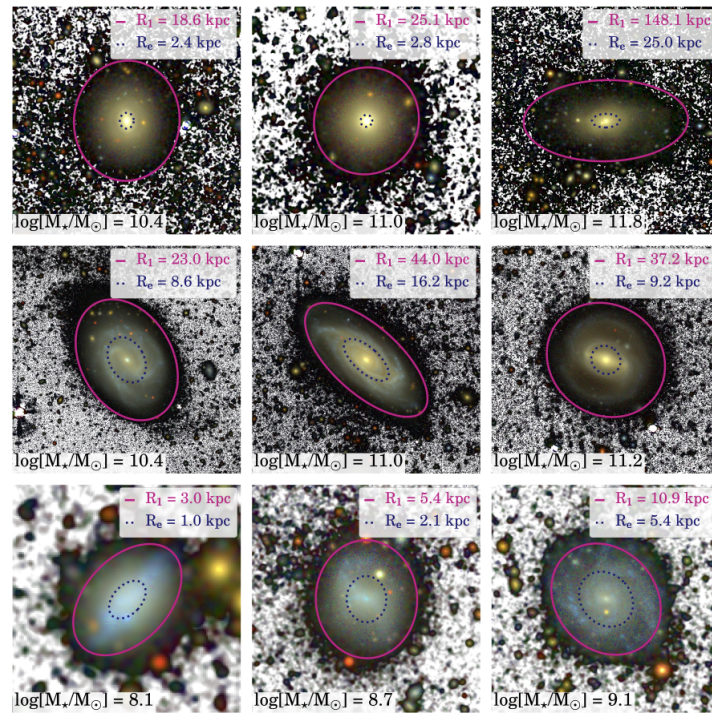


Figura 4.5: Imágenes de galaxias que muestran la ubicación de sus r_e y R_1 (extraído de [33]).

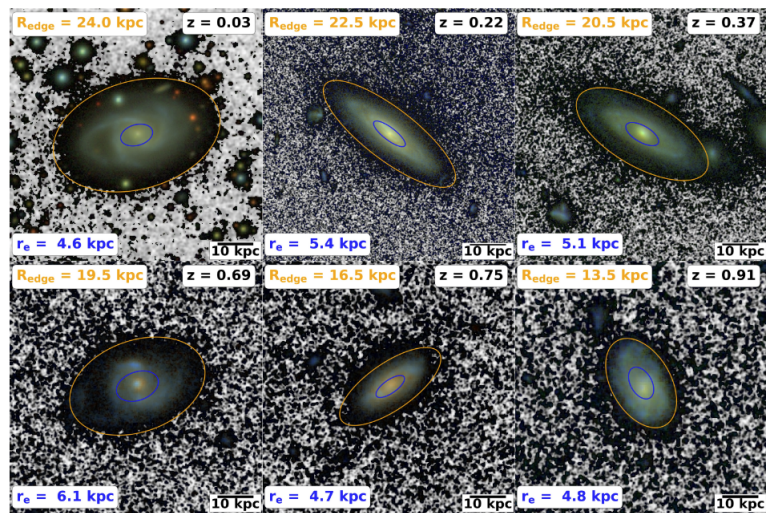


Figura 4.6: Imágenes de galaxias que muestran la ubicación de sus r_e y R_{edge} (extraído de [2]).

5: Metodología experimental

La detección de bordes de galaxias del espacio distante será el objetivo principal de la experimentación llevada a cabo en este capítulo. Primero, se explicará el origen y las características del conjunto de datos del que se parte. Posteriormente, se detallará la relación de experimentos diseñada, siendo por último analizados los resultados obtenidos para cada configuración.

5.1. Conjunto de datos

En esta sección se realizará un estudio del conjunto de datos que ha sido utilizado para realizar la experimentación entorno a la que pivota el trabajo.

5.1.1. Descripción

El conjunto de datos que se ha utilizado en este trabajo ha sido facilitado por el profesor Fernando Buitrago Alonso. Consta de 1052 imágenes provenientes del proyecto *Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey* (CANDELS) [12], el proyecto de mayor magnitud llevado a cabo por el Telescopio Hubble. CANDELS está diseñado para documentar el primer tercio de la evolución galáctica, en el rango aproximado de *redshift* 8-1.5. Cuenta con imágenes de más de 250000 galaxias distantes utilizando tres cámaras distintas del telescopio espacial Hubble, desde el ultravioleta medio hasta el infrarrojo cercano. Los 5 campos de exposición disponibles en el estudio, los cuáles se usarán en el presente trabajo, son: *Great Observatories Origins Deep Survey - North* (GOODS-N), *Great Observatories Origins Deep Survey - South* (GOODS-S), *Ultra Deep Survey* (UDS), *Extended Groth Strip* (EGS) y *Cosmological Evolution Survey* (COSMOS).

Las imágenes capturadas con la cámara de gran angular *Wide Field Camera 3* (WFC3) del telescopio Hubble fueron recortadas con un tamaño fijo de *stamp* de 12x12 arcosegundos, esto es, 200x200 píxeles. Por otro lado, las imágenes provenientes de la *Advanced Camera for Surveys* (ACS) se generaron con una resolución de 400x400 píxeles, aunque posteriormente

se redimensionaron para ajustarse al mismo tamaño que las imágenes capturadas con WFC3.

Como el objetivo del trabajo es el de identificar los límites de las galaxias junto con las imágenes se han proporcionado sus correspondientes máscaras de segmentación, una por imagen. Dichas máscaras están compuestas por píxeles que tienen la misma intensidad cuando forman parte de objetos del mismo tipo. Las máscaras originales contemplan tres posibles cuerpos: ruido, galaxia secundaria y galaxia central (Figura 5.1). Las galaxias de interés en este trabajo son las centrales, por lo que dichas máscaras serán modificadas en consecuencia, tal y como se detallará en la sección 5.1.4.

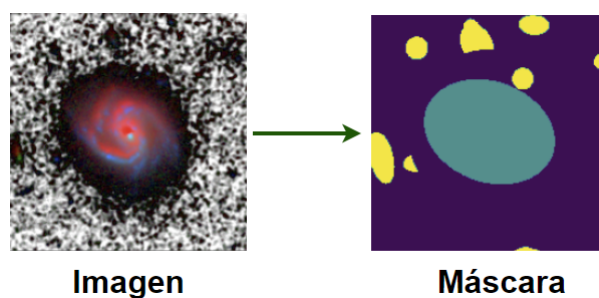


Figura 5.1: Ejemplo de imagen de una galaxia junto con su máscara asociada.

5.1.2. *Astronomic Augmentations*

Contar con 1052 imágenes para un problema tan complejo como el de delimitar los bordes de objetos tremendamente difusos a partir de un *dataset* con una gran cantidad de ruido, debido a la lejanía de los cuerpos capturados, no es todo lo deseable posible. Además, uno de los problemas que surgen de manera más recurrente durante el entrenamiento de redes neuronales convolucionales profundas es el del sobreajuste, provocando capacidades de generalización pobres sobre conjuntos de datos no vistos por los algoritmos. El sobreajuste puede mitigarse incrementando el tamaño del conjunto de datos, entre otras medidas, lo que fuerza a los modelos a generalizar conocimiento y extraer únicamente aquellas características más importantes de los datos usados para ajustar sus pesos.

Por ello, en colaboración con el grupo de investigación bajo el cual se ha desarrollado parte de este trabajo, se desarrolla la novedosa técnica de las *astronomic augmentations*. Dicho concepto, similar al de *data augmentation* en su idea, hace referencia a la capacidad de generar varias imágenes diferentes para una misma galaxia. Al igual que el *data augmentation*, se fundamenta en el concepto de que extrayendo más información del conjunto de datos a partir de transformaciones de los mismos puede llegar a mejorarse el desempeño de los modelos. Una taxonomía de los tipos de *data augmentation* más comunes, adaptada de [29], puede encontrarse en la Figura 5.2. Algunas técnicas de *data augmentation* están relacionadas, ver las flechas de colores, con el correspondiente

esquema de metaaprendizaje, es decir, algoritmos de aprendizaje automático que aprenden a proporcionar transformaciones de los datos óptimas que sirven como entrada a otros algoritmos de *machine learning*.

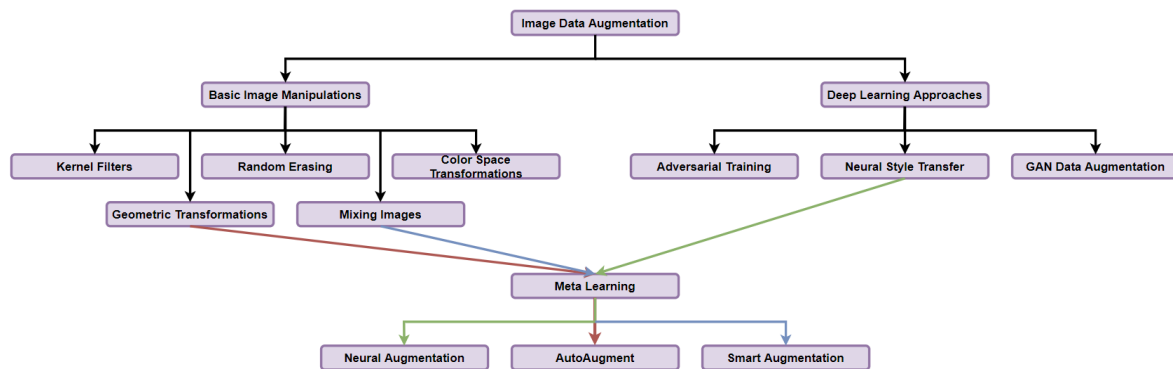


Figura 5.2: Taxonomía de *data augmentations* aplicadas en visión por computador.

Las *astronomic augmentations* que se han utilizado en este trabajo han sido generadas a partir de información que los expertos en astrofísica utilizan para encontrar las truncaciones de las galaxias de manera manual. De esta manera, dichas transformaciones introducen a los posteriores algoritmos de aprendizaje automático valiosas características físicas que definen las propiedades de las galaxias.

Las *astronomic augmentations* utilizadas en este trabajo, denominadas visuales, se han obtenido mediante la combinación de las distintas bandas de observación del espectro electromagnético en las que se capturan las imágenes, manteniendo siempre la restricción de que las bandas o filtros azules, más cercanos al ultravioleta, han de situarse antes que los filtros más cercanos al infrarrojo. De esta manera, y contando con las bandas H, I, J y V, se pueden construir 4 combinaciones de imágenes preservando la restricción comentada: HIV, HJI, HJV, JIV (Figura 5.3). Situando las *astronomic augmentations* en la taxonomía de la Figura 5.2, éstas pueden describirse como transformaciones en el espacio de colores, dentro de las manipulaciones de imágenes que no son realizadas mediante aprendizaje profundo.

En la etapa de experimentación las imágenes se dividirán en dos categorías. La primera de ellas recogerá aquellas que han sido formadas mediante la composición de los canales H, J e I, en ese orden, conformando lo que denominará conjunto de datos base. Por su parte, la segunda categoría incluirá, además de las imágenes base, las otras tres representaciones creadas. Se estudiará si, como es de esperar, la inclusión de *data augmentation* con sentido astronómico provoca una mejoría en los resultados en lo que respecta a los modelos ajustados a partir del conjunto de datos base.

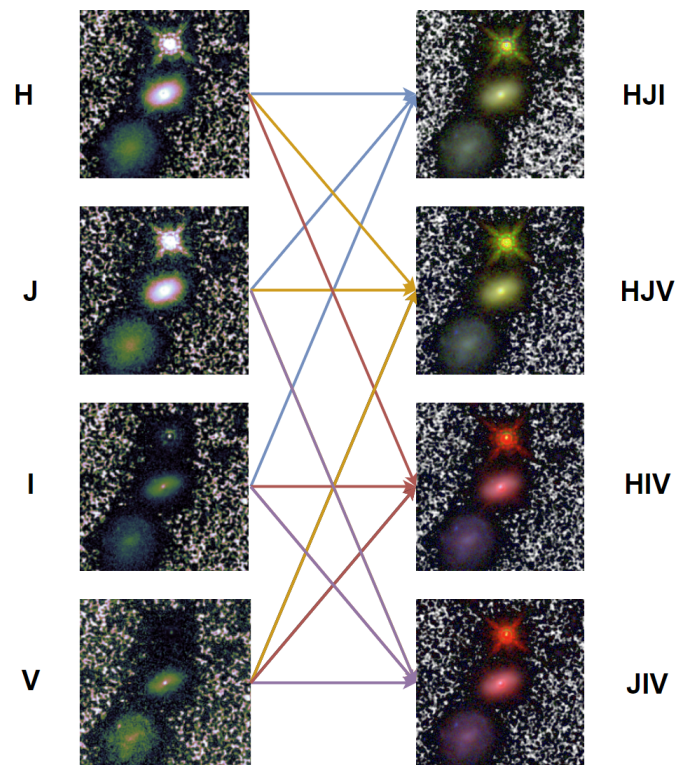


Figura 5.3: Combinación de las bandas H, J, I y V para formar las *astronomic augmentations*.

5.1.3. Exploración

La etapa de exploración de datos se llevará a cabo con las imágenes generadas a partir de la combinación de los espectros electromagnéticos H, J e I, en ese orden, al constituir éstas el conjunto de datos base sin la inclusión de *astronomic augmentations*. Como ya se ha comentado en estadios anteriores del trabajo, dicho conjunto de datos está formado por 1052 imágenes y 1052 máscaras, relacionándose ambas en una correspondencia uno a uno. La proporción de imágenes según el campo de exposición puede apreciarse en la Figura 5.4. La mayor cuota de imágenes es acaparada por el campo [GOODS-N](#) (27.3%), seguido de [COSMOS](#) (24.3%), [EGS](#) (18.7%), [GOODS-S](#) (15.2%) y [UDS](#) (14.4%).

Las imágenes tienen una resolución de 200 x 200 píxeles y una profundidad de 32 bits, es decir, los 3 canales correspondientes a los espectros H, J e I y un cuarto canal de transparencia típicamente conocido como canal *alpha*, el cual define la opacidad de cada píxel en la imagen. Dicho canal toma siempre el máximo valor de transparencia, 255, al ser cada canal de 8 bits. Por su parte, el valor mínimo de las imágenes es de 0 y el máximo de 255.

Hablando de las máscaras, éstas también tienen una resolución de 200 x 200 píxeles. Su profundidad es de 8 bits, por lo que cada máscara puede ser interpretada como una imagen de grises. El valor mínimo que pueden tomar los píxeles de las máscaras es uno, correspondiéndose con las zonas de ruido. El valor máximo posible es tres, correspondiéndose a las zonas de galaxia secundaria. Por su parte, aquellas zonas cuyos píxeles toman el valor de dos representan a la galaxia principal de la imagen asociada a la máscara.

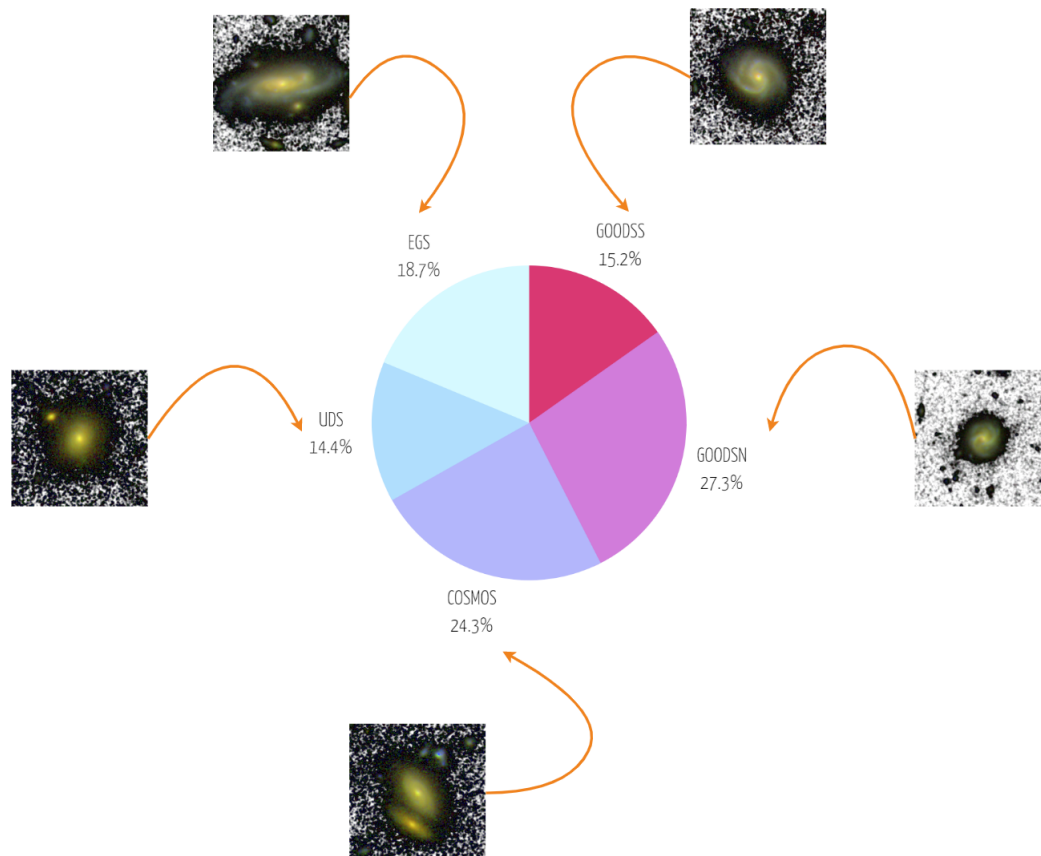


Figura 5.4: Proporción de imágenes según campo de exposición.

La distribución del tamaño de las galaxias según el campo de exposición, calculado a partir de la truncación presente en las máscaras, puede apreciarse en la Figura 5.5, y un gráfico de líneas relacionando el número de imágenes respecto al tamaño de las galaxias en las mismas, igualmente según el campo de exposición, se visualiza en la Figura 5.6. Para complementar, un resumen descriptivo del tamaño de los cuerpos galácticos se encuentra en la Tabla 5.1. Todos los campos de exposición presentan una distribución del tamaño de las galaxias similar. El tamaño medio mayor de cuerpo es alcanzado en el campo de exposición **COSMOS**, con 3127.09 píxeles de superficie, mientras que el menor tamaño medio es proporcionado por el campo **UDS**, con 1817.68 píxeles de superficie. No obstante, atendiendo a la mediana, la cual ofrece unos resultados más robustos debido a su naturaleza, el campo de exposición **GOODSS** es el que presenta galaxias de mayor

Estadístico	GOODSS	GOODSN	COSMOS	UDS	EGS
Media	2606.63	2597.19	3127.09	1817.68	2657.97
Moda	877	307	235	151	939
Mediana	1822.50	1755.00	1799.50	1339.50	1713.00
Mínimo	277	193	151	151	213
Máximo	17513	35434	29096	8713	30093
Desviación típica	2619.20	3732.95	3797.03	1580.92	3504.88

Tabla 5.1: Estadísticos descriptivos de la superficie de las galaxias, en píxeles, según el campo de exposición.

tamaño, mientras que **UDS** sigue siendo el campo que proporciona galaxias de menor tamaño. La desviación típica muestra que el campo de exposición donde más dispersión hay, en lo que respecta al tamaño de las galaxias, es **COSMOS**, mientras que el campo con un tamaño más uniforme es **UDS**. La galaxia más masiva pertenece al campo de exposición **GOODSN**, con una superficie de 35434 píxeles del total 40000, mientras que las galaxias más pequeñas pertenecen a los campos de exposición **COSMOS** y **UDS**, con una superficie de únicamente 151 píxeles. Dichas galaxias pueden apreciarse en la Figura 5.7.

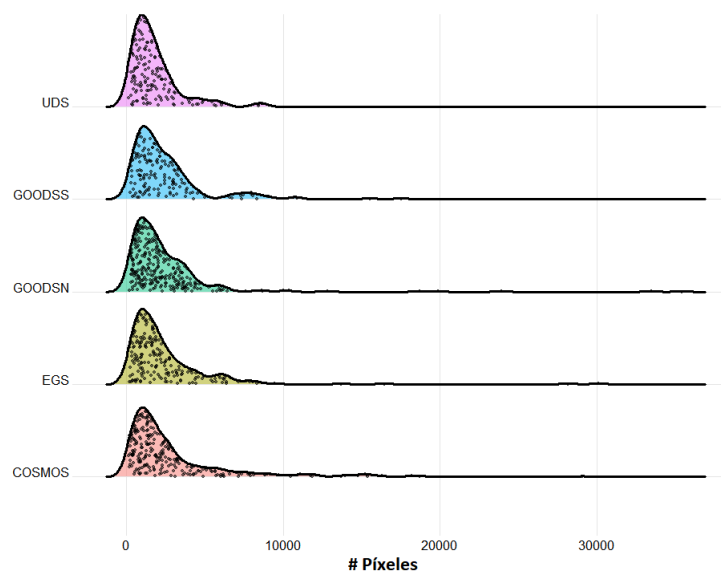


Figura 5.5: Gráfico *ridgeline* mostrando la distribución del tamaño de las galaxias según el campo de exposición.

5.1.4. Transformación

El *pipeline* de transformación de datos que se ha diseñado en este trabajo está compuesto por dos fases. A continuación se detalla cada una de las etapas.

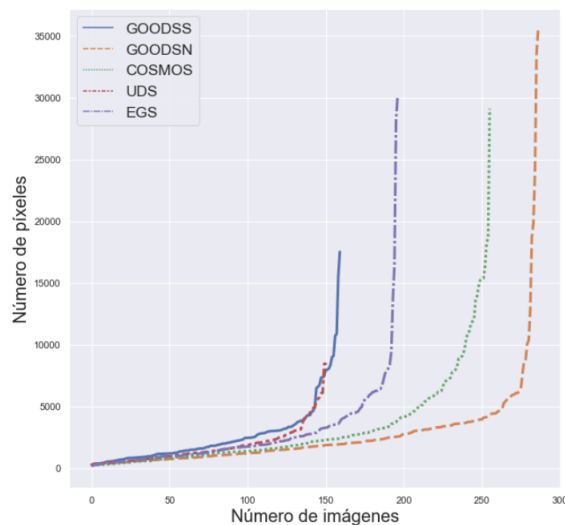


Figura 5.6: Gráfico de líneas mostrando la relación entre el número de imágenes y el tamaño de la galaxia de cada una, ordenadas de menor a mayor superficie, según el campo de exposición.

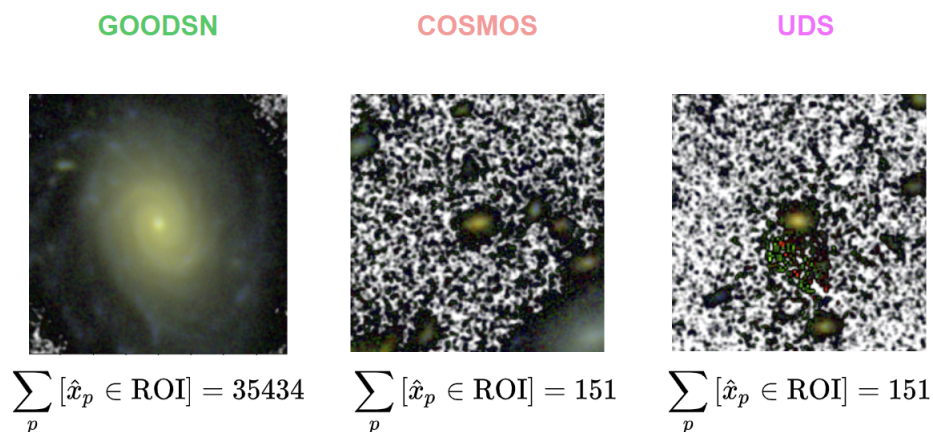


Figura 5.7: Galaxia más masiva y galaxias más pequeñas del conjunto de datos.

5.1.4.1. Eliminación de galaxias secundarias

En esta primera etapa se adaptaron las máscaras para el problema de detectar los bordes del cuerpo galáctico principal de cada imagen, por lo que se eliminó toda aquella información de las máscaras relacionada con la presencia de galaxias secundarias que no fuesen de interés. Recordemos que los píxeles de las máscaras originales pueden tomar tres posibles valores: 1 para las zonas de ruido, 2 para las zonas con galaxia principal y 3 para las zonas con presencia de galaxia secundaria. Por ello, todas las máscaras se modifican para que las zonas con galaxia secundaria tomen el mismo valor que las zonas con ruido, produciendo como resultado máscaras dicotómicas donde cada píxel representa

la ausencia o presencia de galaxia principal en esa zona. El resultado de la transformación se encuentra en la Figura 5.8. Dicha transformación se aplica a las 1052 máscaras.

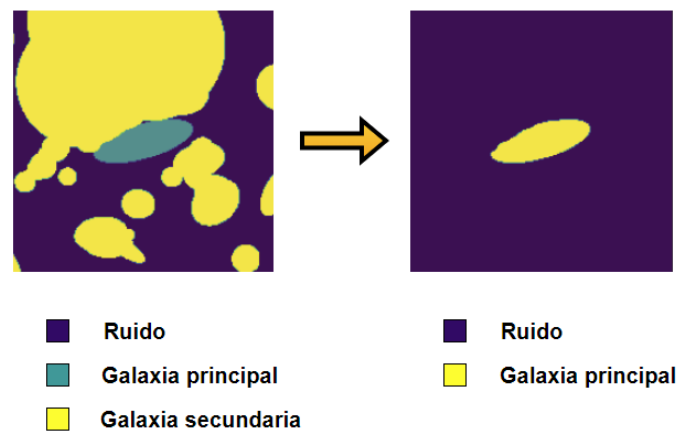


Figura 5.8: Etapa inicial del *pipeline* de transformación.

5.1.4.2. Rellenado de huecos por superposición de cuerpos

La segunda etapa del *pipeline* de transformación viene motivada por la presencia de superposiciones de cuerpos galácticos en la proyección bidimensional del espacio que se recoge en las imágenes. Dicha superposición, cuando ocurre de tal manera que una galaxia se sobrepone a otra galaxia de mayor tamaño, provoca que en las máscaras la zona que ocupa la galaxia pequeña superpuesta no pertenezca a la galaxia principal. No obstante, una galaxia es una estructura continua, sin huecos en su interior, por lo que si dicha superposición es completa cabe esperar que la galaxia original si exista detrás de la galaxia de menor tamaño. De esta manera, ante la ocurrencia de este tipo de situaciones, se modifica el valor de los píxeles de la máscara asociados a las galaxias superpuestas para que tomen el valor correspondiente con la galaxia central de interés. El procedimiento queda descrito gráficamente en la Figura 5.9. Se detectan 20 máscaras a las que hay que aplicarle dicha transformación, para lo cual se hace uso de la biblioteca *software* de visión por computador `OpenCV`, desde su *API* de Python.



Figura 5.9: Etapa secundaria del *pipeline* de transformación.

Concretamente, se realiza una búsqueda de superficies convexas internas, también llamadas *blobs*, a la superficie ocupada por la galaxia principal.

5.2. Diseño experimental

Para encontrar la truncación de las galaxias y tratar de replicar el comportamiento de los astrónomos con la mayor fidelidad posible se diseña un panorama experimental caracterizado por el uso de arquitecturas de aprendizaje profundo en el estado del arte en conjunción con técnicas de entrenamiento punteras. El problema de detección de los límites de las galaxias se aborda como un problema de segmentación semántica, donde el objetivo de la metodología *deep learning* a desarrollar es el de categorizar todos los píxeles de una imagen. La truncación vendrá definida por el perímetro del conjunto de píxeles identificados como galaxia.

5.2.1. Configuraciones arquitectónicas

De esta manera, se utilizarán tres esquemas de construcción de modelos de segmentación semántica: las redes en U o *U-Nets*, las redes en U anidadas o *U-Nets ++* y las redes DeepLab, en su versión tercera. Todos los esquemas han sido cuidadosamente explicados en el tercer capítulo de este trabajo. Los tres tipos de modelos están basados en un enfoque *encoder-decoder*. La primera etapa extrae las características más importante de las entradas, residentes en un espacio p -dimensional, realizando una proyección de las mismas en un espacio latente q -dimensional, con $q < p$ típicamente. El *encoder* está compuesto en su mayoría por operaciones de *downsampling* y convoluciones regulares. La segunda etapa trata de proyectar las características discriminatorias de menor resolución aprendidas por el *encoder* sobre el espacio de píxeles original para obtener una clasificación de cada uno de ellos. El *decoder* está compuesto en su mayoría por operaciones de *upsampling* y convoluciones transpuestas.

Dicha característica común en la arquitectura de los tres esquemas permite que se utilicen distintos tipos de redes neuronales convolucionales como *encoders* de los modelos. En este trabajo, se probarán cuatro familias de redes neuronales convolucionales: ResNet, EfficientNet, DenseNet y DPN. Todas ellas han sido explicadas en el capítulo tercero de este trabajo. De la familia ResNet se utilizan los modelos con 18, 34 y 50 capas ocultas. De la familia EfficientNet se utilizan las versiones de la B1 a la B6, ambas incluidas. De la familia DenseNet se utilizan los modelos 121, 161 y 201. Por último, de la familia de redes de ruta dual se utilizan las versiones 68 y 98. Se diseñan modelos que combinan las 3 familias de redes de segmentación con los 14 *encoders* nombrados, exceptuando la casuística que abarca la combinación de las redes DeepLabV3 con las redes DenseNet, que por razones estructurales no pueden ser integradas. Concretamente, las convoluciones dilatadas no pueden ser utilizadas en una arquitectura DenseNet pura. De esta manera, se diseñan un total de 39 modelos de aprendizaje profundo. Cada modelo será entrenado con dos conjuntos de datos, primero con las imágenes formadas por la combinación de los espectros

H, J e I, en ese orden, y posteriormente con el conjunto de datos producido a partir de las *astronomic augmentations*. De esta manera, se realizarán un total de 78 entrenamientos. El número de parámetros de cada combinación para el caso del experimento base, esto es, sin *astronomic augmentations*, se aprecia en la Figura 5.10.

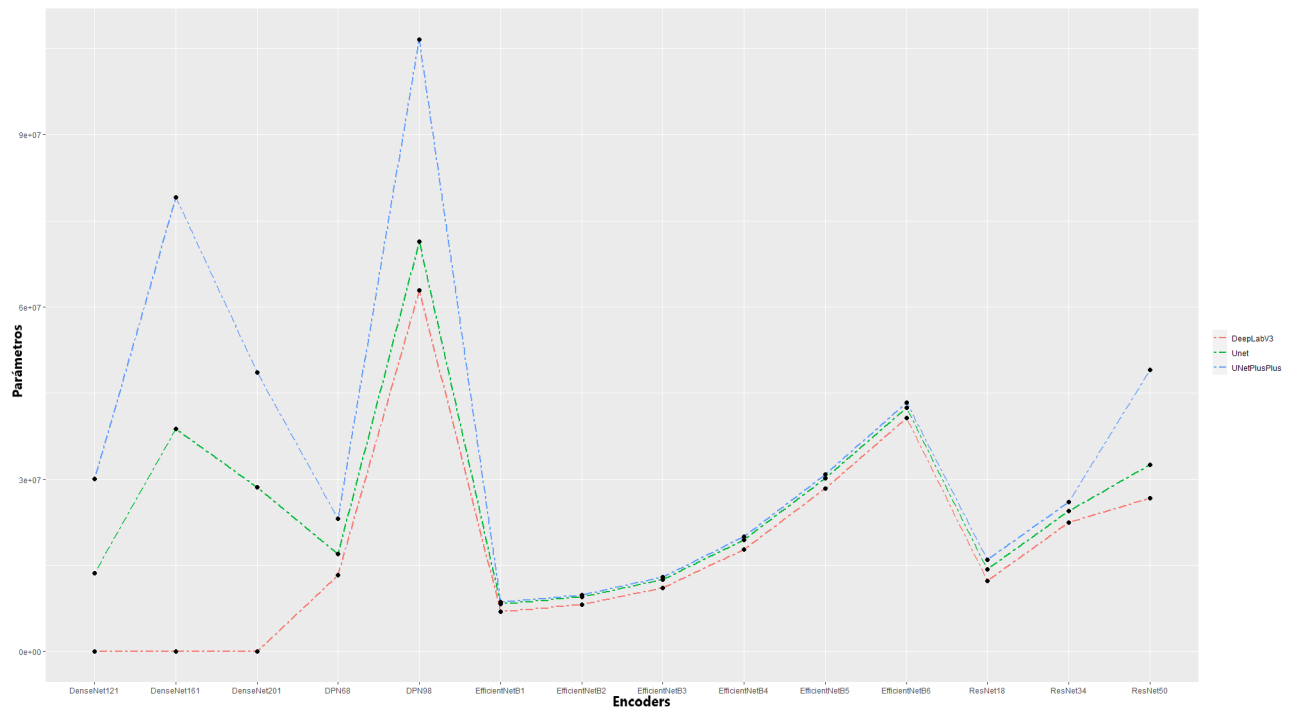


Figura 5.10: Número de parámetros de cada combinación arquitectura de segmentación - *encoder* para el caso del experimento base.

Uno de los conceptos más relevantes que se han tenido en cuenta a la hora de diseñar las redes neuronales es el poder utilizar de manera conjunta las *astronomic augmentations* para realizar la inferencia de una máscara. Típicamente, las redes neuronales de segmentación semántica infieren una máscara a partir de un imagen, caso que ocurre en los experimentos base de este trabajo, donde se cuenta con 1052 imágenes con sus correspondientes 1052 máscaras. No obstante, en el caso de los experimentos visual, se dispone de 4 imágenes por cada máscara segmentación. Parece razonable pensar que la introducción de dichas imágenes a los algoritmos de aprendizaje profundo por individual no tiene mucho sentido, ya que el objetivo es poder aprovechar toda la información suministrada por las *astronomic augmentations*, de manera conjunta, para predecir los bordes de una galaxia.

Teniendo en cuenta dicha necesidad, la primera solución puede pasar por concatenar las imágenes en aquella dimensión de los tensores de entrada que represente la banda electromagnética. De esta manera, se alimentaría a los algoritmos *deep learning* con tensores de 16 filtros o canales, producto de la concatenación de las imágenes compuestas por los espectros HIV, HJI, HJV y JIV. A pesar de que esta solución puede ser perfectamente

válida, en este trabajo se decide explotar el concepto de las convoluciones agrupadas o convoluciones en paralelo. Las convoluciones en paralelo se dieron a conocer principalmente en [19], formando parte de la conocida arquitectura AlexNet (Figura 5.11). En dicha ocasión, este tipo de convoluciones fueron utilizadas para poder entrenar modelos de aprendizaje profundo en tarjetas gráficas con poca memoria RAM. Como detallan los autores del *paper*, la motivación en el uso de convoluciones agrupadas fue la de poder entrenar AlexNet en dos tarjetas gráficas Nvidia GTX 580 con 1.5 GigaBytes de memoria cada una. Dado que el modelo requería algo menos de 3 GigaBytes de RAM para entrenarse, las convoluciones agrupadas permitieron una paralelización del proceso de aprendizaje.

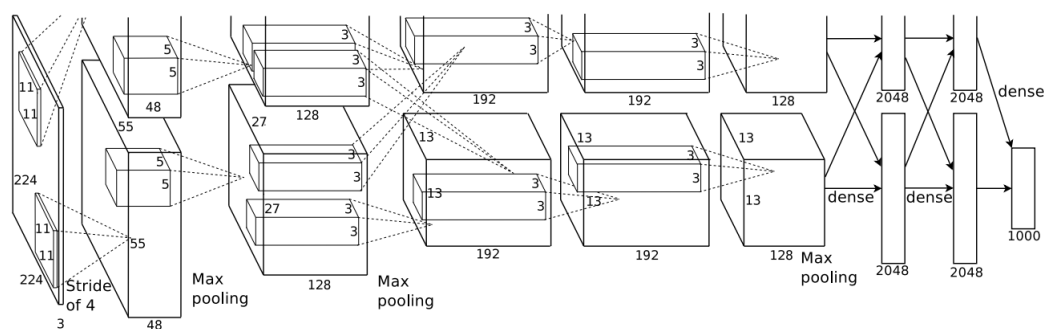


Figura 5.11: Convoluciones agrupadas utilizadas en AlexNet (extraído de [19]).

Sin embargo, en estudios posteriores, se demostró que las convoluciones agrupadas no solo permitían la fragmentación del proceso de entrenamiento de modelos de aprendizaje profundo en distintos dispositivos, si no que su uso conllevaba una clara ventaja, y es que las convoluciones agrupadas parecían aprender una mejor representación de los datos que las convoluciones clásicas. Dicha afirmación queda sustentada por la alta correlación que se produce entre los diferentes filtros de una capa cuando estos son aplicados a un mismo *input*, aunque los resultados que produzca la aplicación de los filtros sean visualmente diferentes. Las convoluciones agrupadas ayudan a que los filtros y mapas de características aprendidos en cada ramificación sean mucho más independientes de sus homólogos paralelos, induciendo al aprendizaje de conceptos que, posteriormente, serán debidamente combinados.

Esta idea fue aprovechada por los autores de las redes ResNext, presentadas en [34], los cuales modificaron los bloques residuales característicos de las arquitecturas ResNet diversificando las entradas al bloque en varios subbloques de la misma topología mediante el uso de convoluciones agrupadas (Figura 5.12). La construcción, que aúna los conceptos de las redes ResNet [13] y los bloques Inception [31], conduce a la creación de redes más amplias que consiguen aprender un conjunto variado de características de bajo y alto nivel.

En este trabajo, se deciden utilizar convoluciones agrupadas como paso previo a la introducción de las distintas *astronomic augmentations* a las redes de segmentación. La

idea es extraer las características primarias más importantes de cada imagen de manera independiente a las demás, suceso que no se puede dar si todas las *astronomic augmentations* se concatenan a lo largo de la dimensión del espectro electromagnético. Un concepto similar es realizado en [30], donde se utilizan convoluciones agrupadas sobre distintas entradas para extraer la información espacial más relevante de cada *input* y posteriormente utilizar una segunda red para, a partir de toda esa información, establecer una categorización del objeto representado en el volumen original.

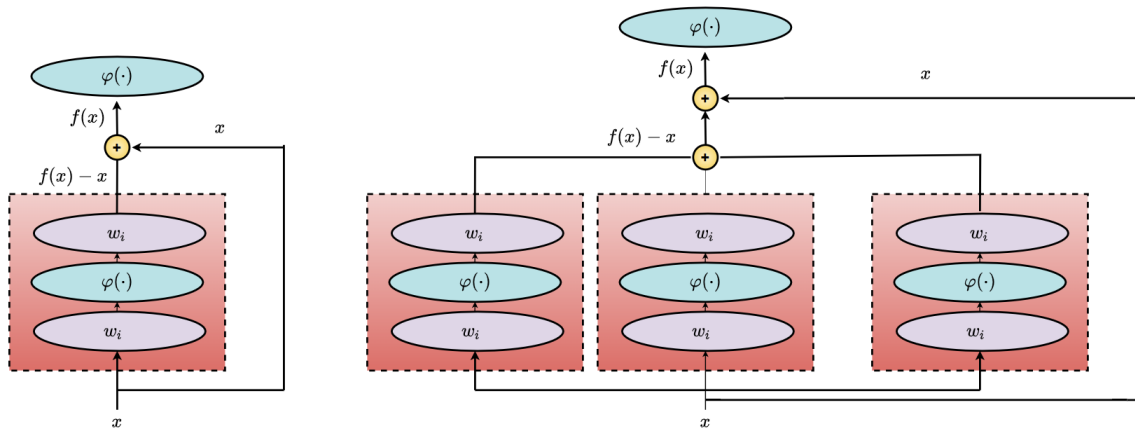


Figura 5.12: Bloque ResNet (izquierda) frente a bloque ResNext (derecha).

De esta manera, un esquema visual de la arquitectura diseñada para realizar el entrenamiento con el *dataset* compuesto por las *astronomic augmentations*, utilizando como modelo de segmentación una red en U anidada, puede ser visto en la Figura 5.13. Cada imagen inicial es introducida en una subred compuesta por dos bloques secuenciales, estando cada bloque formado por una operación de convolución, una normalización de *batch* y una activación no lineal. Dicha subred se replica de manera paralela para los cuatro *inputs* del conjunto de datos visual. El número de filtros en los que se encuentra cada imagen aumenta de 4 a 9, momento en el cual los cuatro volúmenes son concatenados en la dimensión que originalmente representaba el espectro electromagnético, conformando un volumen de la misma altura y anchura que el original pero con una profundidad de 36 canales, en vez de los 16 canales que se obtendrían si se combinaran directamente las imágenes originales. Dicho *output* de la primera etapa es introducido en una red de segmentación, la cual se encargará de delimitar los límites del cuerpo galáctico representado en el volumen. Para igualar las condiciones experimentales, y determinar con mayor peso si la introducción de las *astronomic augmentations* supone una mejoría con respecto al experimento base, se decide que las redes para los experimentos base también estén formadas por la concatenación de dos fases, una primera en la que se aplica la subred inicial a la imagen en los filtros H, J e I y una segunda en la que la imagen en los 9 canales resultantes es tomada por el modelo de segmentación semántica.

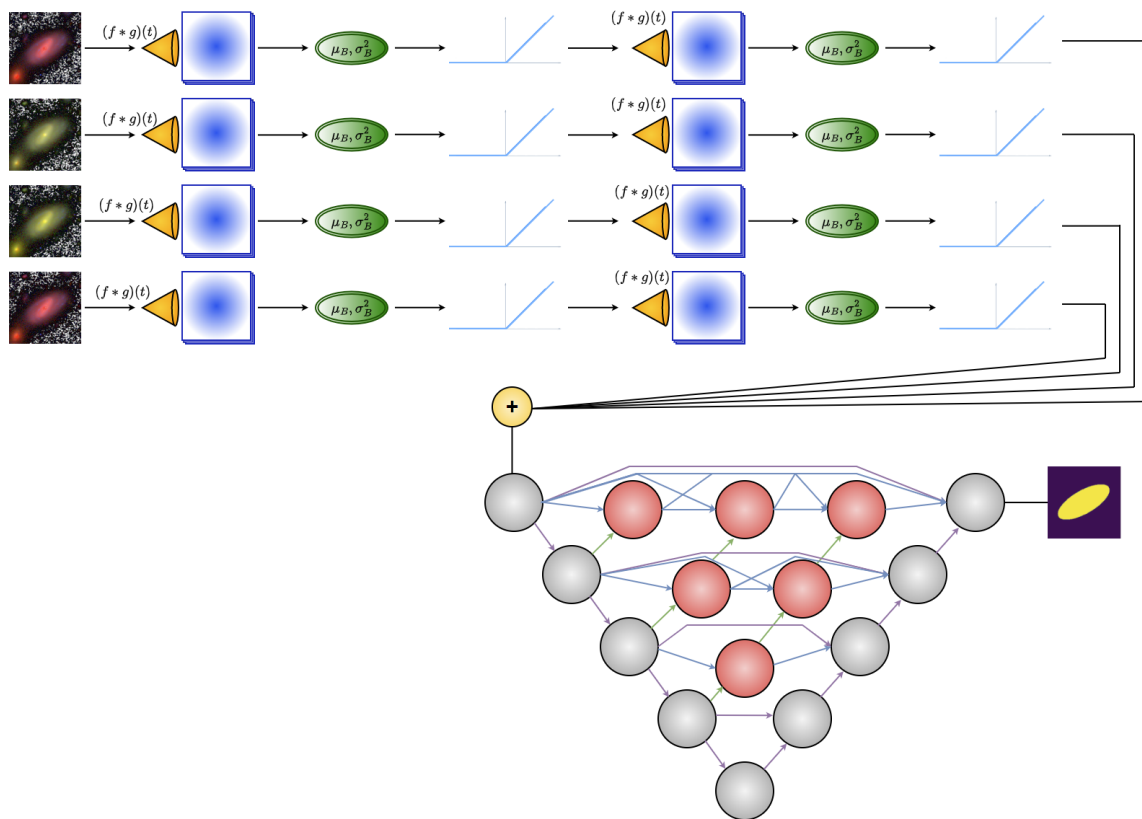


Figura 5.13: Arquitectura diseñada para los experimentos con *astronomic augmentations* utilizando como red de segmentación una U-Net ++.

5.2.2 Hiperparámetros y características del entrenamiento

Este apartado trata los hiperparámetros y decisiones que caracterizan el entrenamiento de las redes neuronales diseñadas en este trabajo. Para empezar, el tamaño de *batch* elegido es de 32 instancias. Dicho hiperparámetro tiene una relación directa con la cantidad de memoria de la que dispone la tarjeta gráfica, y se eligió un tamaño que permitiese el entrenamiento simultáneo de dos modelos de aprendizaje profundo en un dispositivo Nvidia RTX A5000 con 24 GigaBytes de memoria RAM. El optimizador utilizado para ajustar iterativamente los pesos de las redes ha sido ADAM, debido a su conocida solvencia y genial desempeño en una amplia gama de problemas del aprendizaje profundo. La tasa de aprendizaje se mantiene constante durante todos los entrenamientos y se fija en 10^{-3} .

Como función de pérdida se usa la entropía cruzada binaria, la cual puede ser vista como un caso particular de la entropía cruzada. Debido a que las clases se encuentran muy desbalanceadas en las máscaras, es decir, la mayoría de los píxeles pertenecen a zonas de ruido y no a zonas de galaxias, se decide aplicar una ponderación w_i a la función de entropía cruzada. De esta manera, se ponderan más los aciertos y fallos en píxeles pertenecientes a las galaxias que en los píxeles pertenecientes al ruido, al ser éstos últimos

mucho más mayoritarios y, por tanto, correr el riesgo de que la red prediga que todos los píxeles de la imagen son ruido. El peso que se le da a la clase que representa las galaxias es de 0.925, mientras que el peso que se le da a los píxeles que representan ruido es de 0.075. Dicha ponderación se ha asignado de manera inversa a la proporción de píxeles de galaxia y ruido disponibles a lo largo de todas las imágenes del conjunto de datos. De esta manera, la función de pérdida utilizada es:

$$J_{\text{wbce}} = -\frac{1}{M} \sum_{m=1}^M [w \cdot y_m \cdot \log(h_{\theta}(x_m)) + (1 - y_m) \cdot \log(1 - h_{\theta}(x_m))]$$

donde M es el número de observaciones, w el peso asignado, y_m la etiqueta del ejemplo m , x_m el ejemplo m y h_{θ} el modelo con los pesos θ . De esta manera, a la clase minoritaria, es decir las galaxias, se le asigna un mayor peso que a la clase mayoritaria, es decir, el ruido.

El número de épocas utilizadas para entrenar los modelos es de 1000, y se utilizan principalmente 3 técnicas de regularización con el objetivo de prevenir la memorización de los ejemplos de entrenamiento, incurriendo en sobreajuste, y lograr una mejor generalización del conocimiento aprendido sobre observaciones no vistas por la red. La primera técnica utilizada es la transformación artificial aleatoria de las imágenes en cada época mediante la técnica conocida como *data augmentation*. Cada imagen, antes de ser introducida a la red, es rotada 0° , 90° , 180° o 270° de forma aleatoria y, una vez aplicada dicha transformación, la imagen es mantenida sin más cambios, espejada entorno al eje de abscisas o espejada respecto al eje de ordenadas de manera aleatoria. De esta manera, hay 11 posibles combinaciones que inducen a imágenes diferentes de la original, provocando que a la red le sea mucho más complicado memorizar los ejemplos de entrenamiento. En la Figura 5.14 se aprecia un ejemplo del *pipeline* de *data augmentation* diseñado.

La segunda de las técnicas de *data augmentation* utilizada ha sido la adición de ruido aleatorio, proveniente de una distribución normal, en cada época a cada ejemplo de entrenamiento. El ruido añadido a cada imagen proviene de una distribución normal con media 0 y desviación típica 0.01, 0.05 o 0.1, elegido dicho valor de manera aleatoria para cada imagen. Por último, se añade regularización Ridge, también conocida como l_2 o regularización de Tikhonov. Esta técnica, originalmente propuesta en [15], impone una penalización a los coeficientes estimados en un problema de regresión lineal, limitando la flexibilidad de la relación entre los predictores del problema y la variable objetivo. De esta manera, la expresión de la suma de residuales al cuadrado con penalización Ridge es:

$$RSS_{\text{Ridge}} = \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \sum_{j=1}^p \beta_j^2$$

donde λ es un parámetro que controla el grado de penalización. Cuanto más grande sea éste, menores serán los coeficientes estimados y menos flexible será el ajuste proporcionado. Trasladando el concepto al aprendizaje profundo, y en el contexto de este trabajo, la

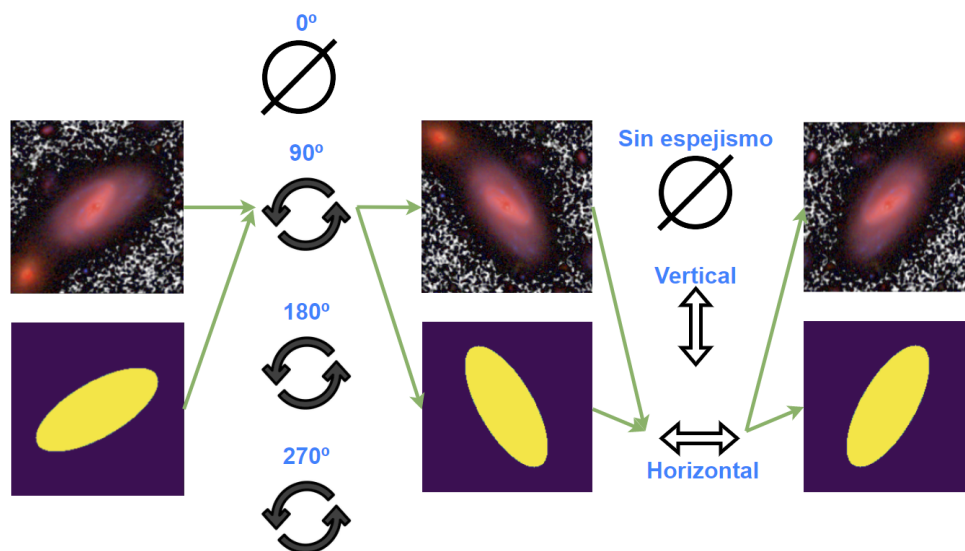


Figura 5.14: *Data augmentation* aplicado a la muestra de entrenamiento.

penalización es añadida a los pesos del modelo:

$$L(y, h_{\theta}(y)) = \sum_{i=1}^n J_{\text{WBCE}}(y_i, h_{\theta}(y_i)) + \lambda \sum_{j=1}^p w_j^2$$

siendo w_j el peso j -ésimo del modelo en cuestión y p el número total de pesos del modelo.

El modo de evaluar los algoritmos *deep learning* y estimar la tasa de error real cometida es mediante la técnica *hold-out*, donde una parte del *dataset* se reserva como conjunto de entrenamiento para ajustar los pesos del modelo y otra parte del como conjunto de test para evaluar el rendimiento del mismo. El número de instancias que componen el conjunto de entrenamiento es de 952, y el número de instancias reservadas para el conjunto de test es de 100. Por su parte, el conjunto de entrenamiento es dividido en 2 subconjuntos, entrenamiento y validación, en una proporción del 80 % y el 20 % respectivamente. Mientras que el conjunto de entrenamiento se utilizará para llevar a cabo el proceso de aprendizaje de los algoritmos *deep learning* el conjunto de validación se utilizará para monitorizar el desempeño de la red sobre un subconjunto del *dataset* no visto y decidir en qué momento del entrenamiento los pesos de la red inducen a un mejor comportamiento cuando las redes se aplican sobre instancias nuevas.

Para evaluar los modelos se utilizan 4 métricas, la tasa de acierto, el *precision*, el *recall* y el coeficiente de Sørensen–Dice. La tasa de acierto, calculada como el número de píxeles bien clasificados dividido entre el número total de píxeles, tiene un problema: no es sensible al coste de la clasificación incorrecta. Por ello, es preferible utilizar alguna métrica más compleja, que tenga en cuenta el coste de los fallos y se adapte al desbalanceo

presente en las imágenes. La métrica elegida es el coeficiente de Sørensen–Dice. Dicha métrica representa la media armónica entre el *precision*, número de píxeles correctamente predichos como galaxia de entre el total de píxeles predichos como galaxia, y el *recall*, número de píxeles correctamente predichos como galaxia de entre el total de píxeles reales de la galaxia. Un *precision* perfecto podría lograrse con la predicción correcta de un único píxel como galaxia, prediciendo el resto de píxeles como ruido. Un *recall* perfecto podría lograrse con la predicción de todos los píxeles de la imagen como galaxia. Por ello, es necesario realizar una ponderación de ambas métricas. La media armónica, al ser un estadístico más exigente y pesimista que sus homólogas aritmética y geométrica, es la utilizada para conformar el coeficiente de Sørensen–Dice. El funcionamiento de la tasa de acierto, el *precision*, el *recall* y el coeficiente de Sørensen–Dice pueden ser apreciados en la Figura 5.15. Además, la métrica es razonablemente robusta a los cambios de escala del objeto de interés al poder ser vista como una medida del *overlap*.



Figura 5.15: Comportamiento del índice Sørensen–Dice frente a la tasa de acierto, el *precision* y el *recall*.

6: Resultados

Este capítulo aborda el análisis y la discusión de los resultados obtenidos tras la ejecución del diseño experimental detallado en el capítulo anterior. Primero, se estudiará el proceso de aprendizaje que han llevado a cabo las diferentes configuraciones. Posteriormente, se presentarán los resultados obtenidos sobre el conjunto de test y, por último, se examinará el efecto que causan las *astronomic augmentations*, se analizará el funcionamiento de las convoluciones agrupadas y se cuantificará detalladamente el desempeño del mejor modelo.

6.1. Evolución del aprendizaje

Analizar el proceso de aprendizaje de los algoritmos *deep learning* puede ser una de las técnicas más efectivas para determinar la ocurrencia de posibles fenómenos como el sobreajuste y el infraajuste, los cuales provocan una gran pérdida de efectividad y rendimiento de los modelos. Con este fin, se estudiará la evolución de la función de pérdida sobre los conjuntos de entrenamiento y validación para las configuraciones entrenadas tanto sobre el *dataset* base como sobre el *dataset* con *astronomic augmentations* o visual.

La evolución de los procesos de aprendizaje para los experimentados llevados a cabo con la arquitectura *U-Net* sobre los conjuntos de datos base y visual se encuentra en las Figuras 6.1 y 6.2 respectivamente. Por lo general, analizando los 28 experimentos, no se aprecia ningún caso en el que haya un marcado sobreajuste, situación en la que se observaría una minimización constante y prolongada de los valores de la función de pérdida sobre el conjunto de entrenamiento que no se vería reflejada sobre el conjunto de validación, donde los valores se estancarían en una asíntota horizontal o incluso aumentarían a partir de cierto punto del proceso de aprendizaje. Tampoco parece que haya diferencias significativas entre la evolución de los procesos de aprendizaje efectuados sobre los conjuntos de datos base y visual. Los valores son razonablemente estables, sin producirse por tanto grandes cambios en las magnitudes del gradiente que induzcan a cambios de tendencia repentinos en las evoluciones del *loss* a lo largo de las épocas.

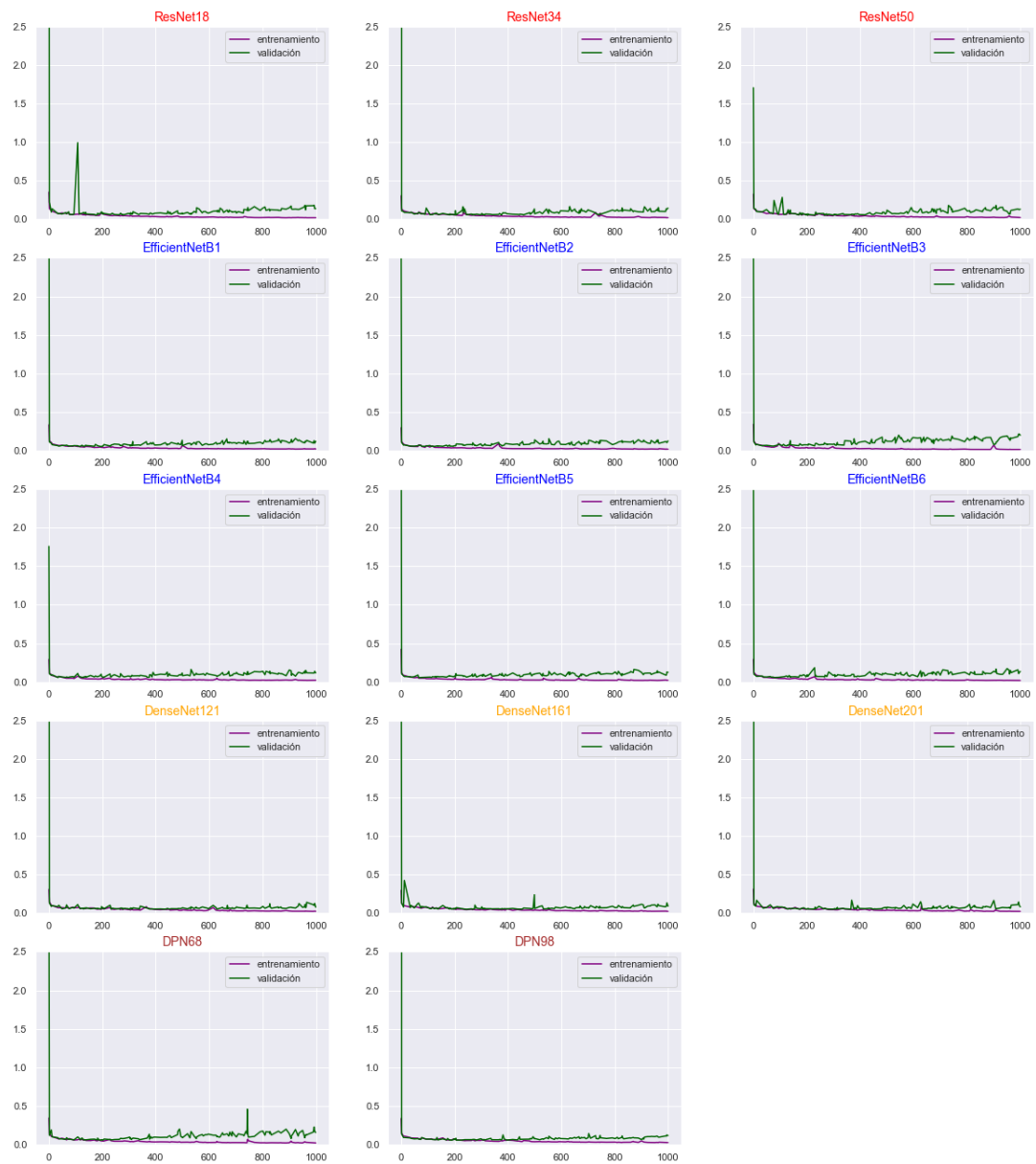


Figura 6.1: Evolución del *loss* sobre los conjuntos de entrenamiento y validación del experimento base para los distintos *encoders* que se incrustan en una arquitectura de tipo *U-Net*.

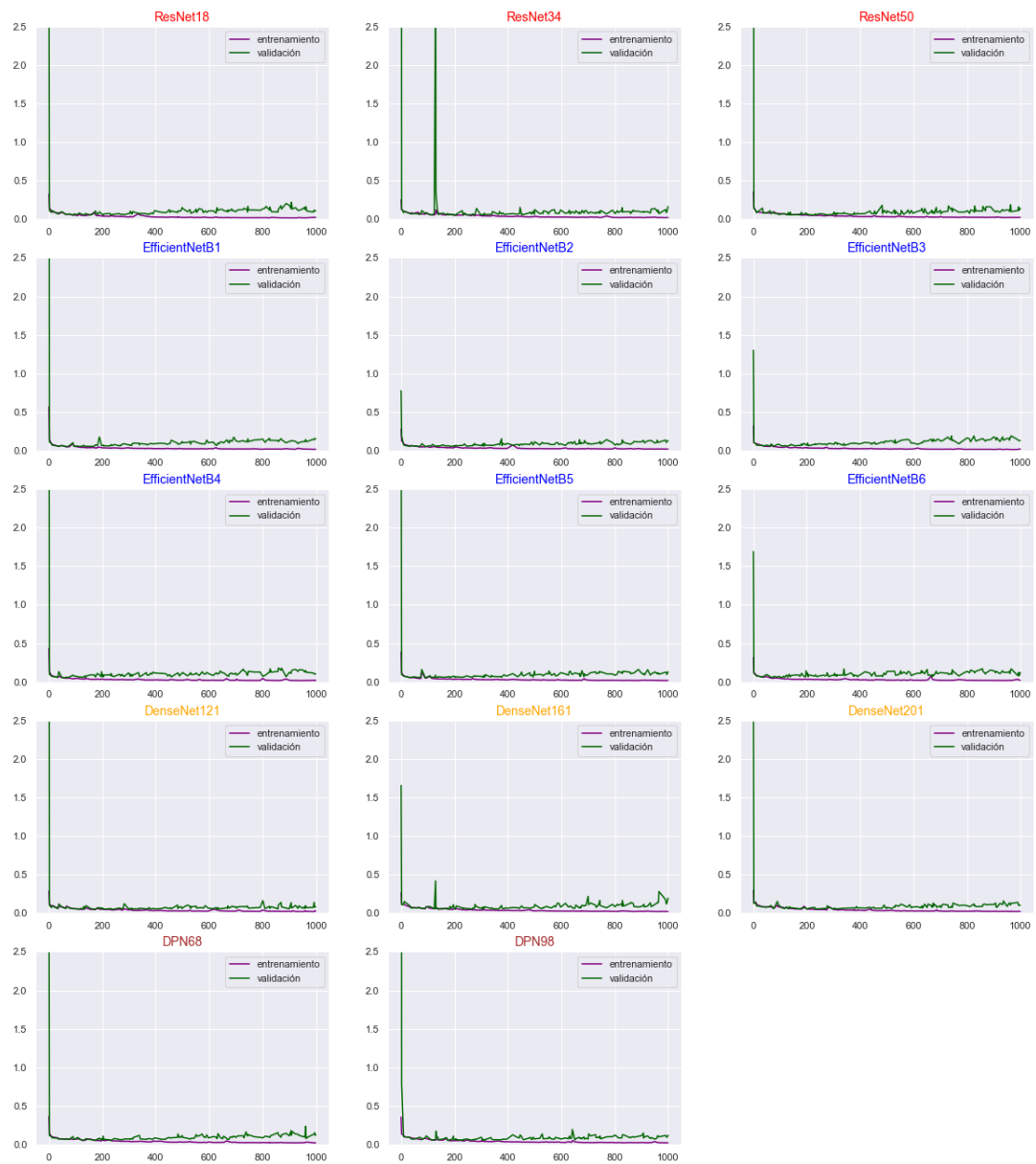


Figura 6.2: Evolución del *loss* sobre los conjuntos de entrenamiento y validación del experimento visual para los distintos *encoders* que se incrustan en una arquitectura de tipo *U-Net*.

De manera similar, la evolución de los procesos de aprendizaje para los experimentos llevados a cabo con la arquitectura *U-Net ++* sobre los conjuntos de datos base y visual (Figuras 6.3 y 6.4) muestra unos patrones parecidos al caso de la *U-Net*. No hay indicios que apunten a un posible sobreajuste o infraajuste de los modelos, por lo que parece que el balance entre la complejidad de los modelos y las técnicas de regularización utilizadas es el óptimo para abordar la dificultad intrínseca en el problema. Quizás, las configuraciones que incluyen a los *encoders* DenseNet161, DenseNet201, DPN68 y DPN98 presentan una oscilación más fuerte en los valores del *loss* sobre el conjunto de validación. No obstante, la situación no es problemática, ya que rápidamente estos tienden a estabilizarse dentro de unos límites razonables, por lo que las oscilaciones puntuales puede que únicamente sean la consecuencia de un proceso de optimización de los pesos ligeramente más inestable. Por último, y mostrando patrones de aprendizaje adecuados tal y como ha sucedido con las dos arquitecturas de segmentación anteriores, en las Figuras 6.5 y 6.6 se representa la evolución de los procesos de aprendizaje para los experimentos llevados a cabo con la arquitectura DeepLabV3 sobre los conjuntos de datos base y visual respectivamente. Quizás, de entre todas las arquitecturas de segmentación estudiadas, esta sea la que presente un comportamiento más inestable. Por un lado, las arquitecturas B1, B2 y B3 de la familia EfficientNet presentan un leve sobreajuste, tanto en las configuraciones entrenadas sobre el conjunto de datos base como en las configuraciones entrenadas sobre el conjunto de datos que incluye las *astronomic augmentations*. Este fenómeno se aprecia debido a que la pérdida sobre el conjunto de validación parece aumentar de manera clara a medida que las épocas se suceden, mientras que la pérdida sobre el conjunto de entrenamiento se mantiene cercana al 0. De igual manera, algunos *encoders*, en conjunción con el *dataset* visual, presentan unas oscilaciones de la pérdida sobre validación puntuales pero muy marcadas. Dichos *encoders*, concretamente EfficientNetB4 y DPN98, no son los 2 de mayor tamaño, ni los 2 de menor tamaño, y ni siquiera pertenecen a la misma familia, por lo que no parece que se puedan extrapolar causas atribuibles a dichos patrones de comportamiento y puede que se trate de una puntualidad que ha ocurrido en los procesos de aprendizaje.

En resumen, parece que, en líneas generales, la arquitectura de segmentación semántica DeepLabV3 induce unos entrenamientos ligeramente más inestables que los efectuados con las arquitecturas *U-Net* y *U-Net++*. De igual manera, se aprecia un ligero sobreajuste en algunas configuraciones de la red que hace uso de las convoluciones dilatadas. No obstante, la tónica general apreciada en el global de los entrenamientos es muy positiva. No parece haber problemas de infraajuste, por lo que las combinaciones de arquitectura-*encoder* elegidas son lo suficientemente complejas como para hacer frente al problema de delimitar los bordes de las galaxias. Tampoco se aprecian fenómenos de sobreajuste generalizados, concluyendo que las técnicas de regularización diseñadas, basadas en la aplicación de transformaciones geométricas, en la adición de ruido y en la penalización Ridge, son efectivas ante la aparición del sobreentrenamiento. Además, la realización del *callback* que selecciona el modelo que maximiza su rendimiento sobre el conjunto de validación añade otra medida de robustez frente a posibles fenómenos de sobreajuste y volatilidad en la evolución de las métricas debido a una optimización compleja de resolver.



Figura 6.3: Evolución del *loss* sobre los conjuntos de entrenamiento y validación del experimento base para los distintos *encoders* que se incrustan en una arquitectura de tipo *U-Net ++*.

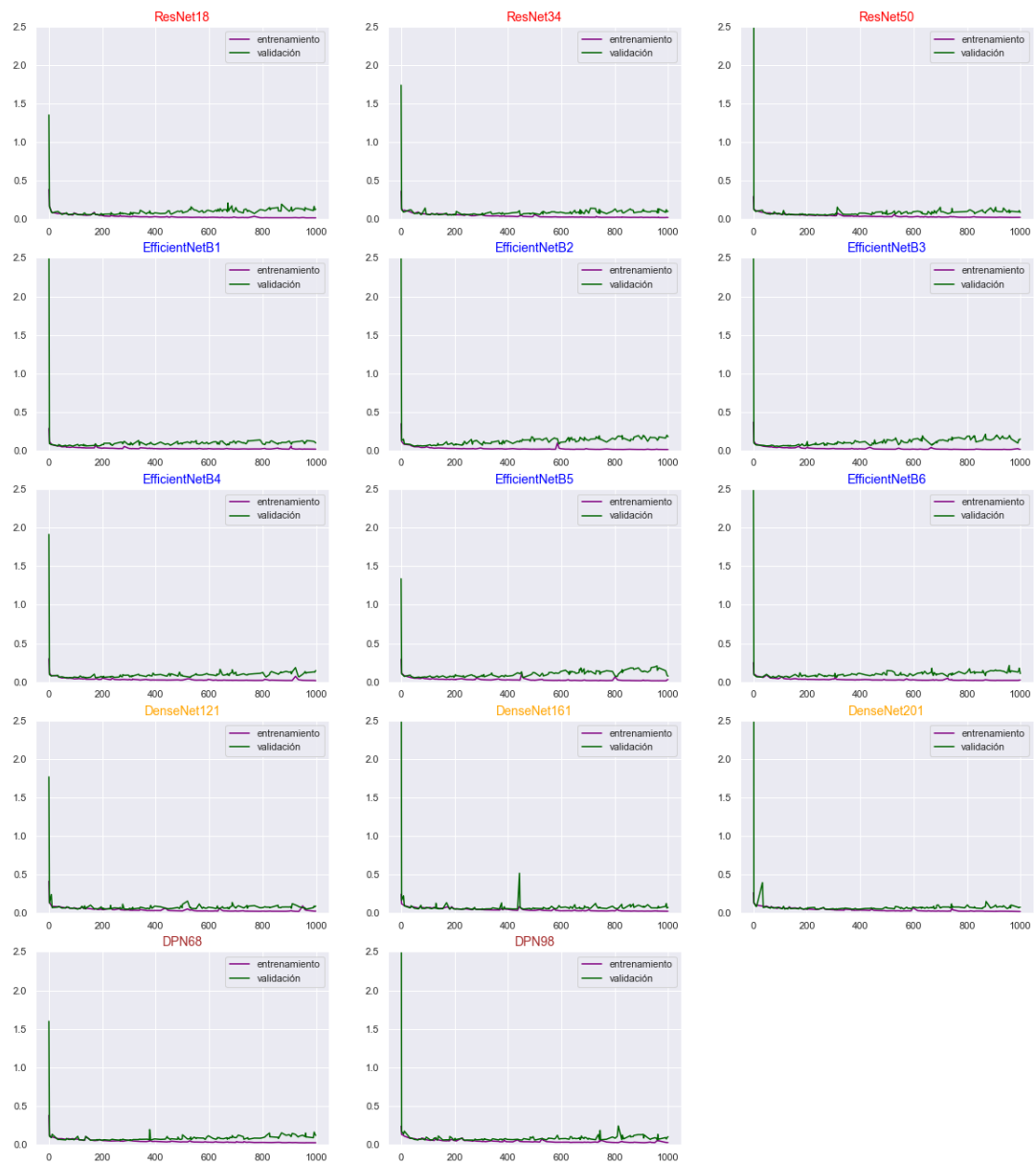


Figura 6.4: Evolución del *loss* sobre los conjuntos de entrenamiento y validación del experimento visual para los distintos *encoders* que se incrustan en una arquitectura de tipo U-Net ++.

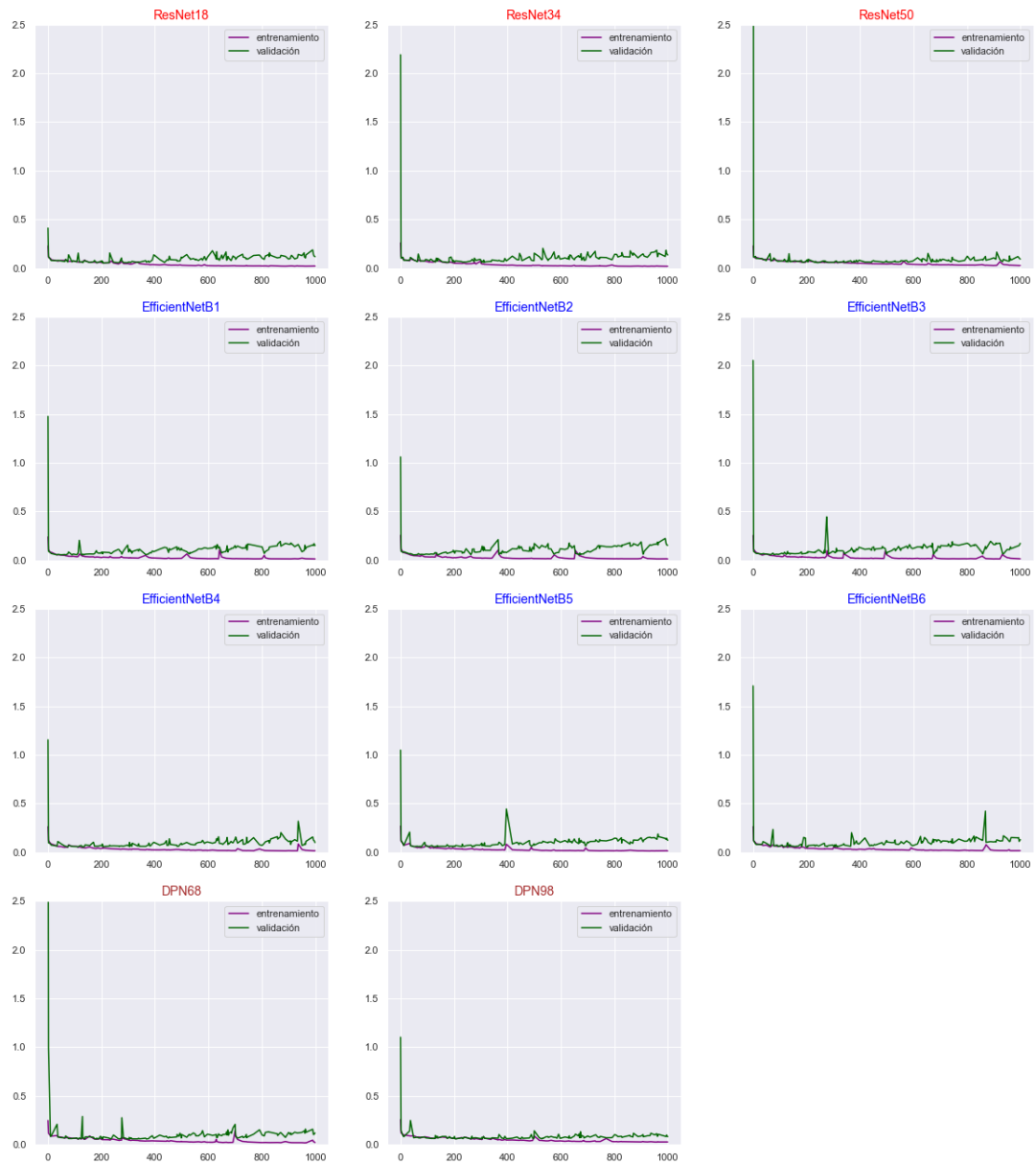


Figura 6.5: Evolución del *loss* sobre los conjuntos de entrenamiento y validación del experimento base para los distintos *encoders* que se incrustan en una arquitectura de tipo DeepLabV3.

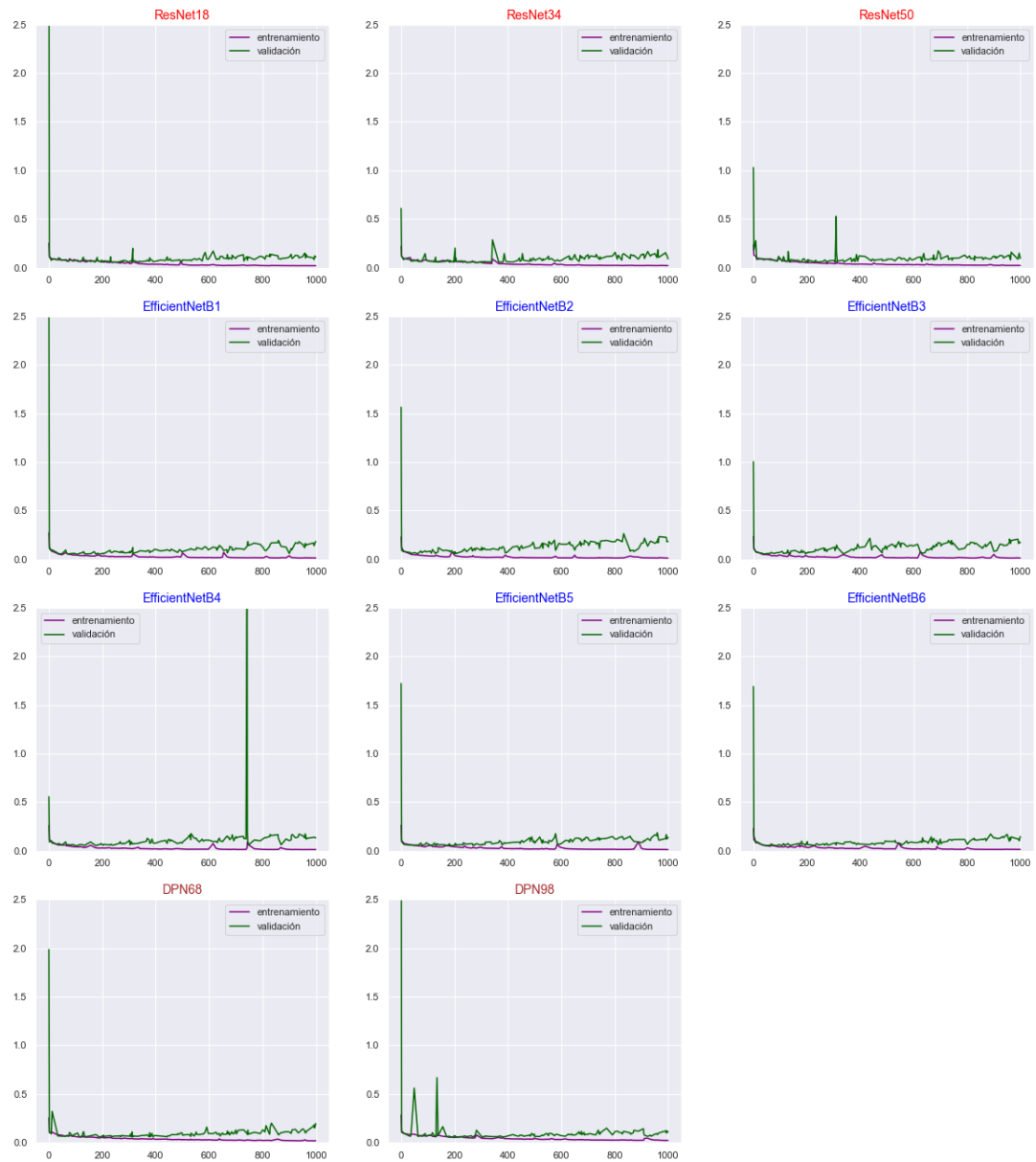


Figura 6.6: Evolución del *loss* sobre los conjuntos de entrenamiento y validación del experimento visual para los distintos *encoders* que se incrustan en una arquitectura de tipo DeepLabV3.

6.2. Resultados sobre el conjunto de test

Una vez analizadas las evoluciones de los procesos de aprendizaje llevados a cabo por los distintos experimentos realizados es el turno de estudiar el rendimiento de cada configuración sobre el conjunto de test. Recordemos que el método de estimación del desempeño real cometido por cada modelo es *hold out*, por lo que las métricas obtenidas sobre el conjunto de test se aproximan al comportamiento real de los algoritmos en el caso de que estuvieran desplegados en un entorno de producción. En este trabajo, el coeficiente de Sørensen–Dice medio será el valor que resumirá el rendimiento de las configuraciones sobre el conjunto de imágenes no vistas.

En la Tabla 6.1 se aprecia el coeficiente de Sørensen–Dice medio de cada experimento sobre el conjunto de test. Resaltados en rojo se encuentran los valores de las 5 configuraciones que inducen un peor rendimiento sobre el conjunto de test. De manera complementaria, de color lima se encuentran marcados los 5 valores obtenidos por las 5 configuraciones con un mejor desempeño promedio sobre el conjunto de test. En término general, los resultados son excelentes. Todas las configuraciones obtienen valores promedios del índice Dice no lejanos al 90 % sobre el conjunto de test, lo cual muestra que la calidad del segmentado y, por tanto, de la detección de bordes de galaxias, es muy buena. El índice Dice, al representar una media armónica entre las métricas *precision* y *recall*, requiere de valores de, como mínimo, entorno al 90 % en ambas métricas para llegar a las cuotas del 90 % que prácticamente se alcanzan por la totalidad de los experimentos. A primera vista, parece que los *encoders* de las familias DenseNet, EfficientNet y DPN logran configuraciones más óptimas que los de la familia ResNet. También parece que las arquitecturas U-Net ++ y DeepLabV3 logran resultados más prometedores que la U-Net.

Bajo una primera aproximación en el estudio del efecto de las *astronomic augmentations*, de los 5 mejores índices Dice obtenidos 3 de ellos son producto del entrenamiento de modelos con el conjunto de datos visual, mientras que 2 de ellos son obtenidos a partir de procesos de aprendizaje efectuados con el conjunto de datos base, es decir, sin las *astronomic augmentations*. Por su parte, de los 5 peores índices Dice obtenidos, 3 de ellos son producto del entrenamiento de modelos con el conjunto de datos base, mientras que los otros 2 son obtenidos a partir de procesos de aprendizaje efectuados con el conjunto de datos visual. Dichos resultados provocan una intuición acerca de que la inclusión del *data augmentation* con sentido astronómico puede ser de gran ayuda para mejorar la calidad en la tarea de detección de galaxias de los modelos de aprendizaje profundo.

6.3. Análisis del efecto *astronomic augmentations*

Escudriñando con más atención los resultados obtenidos, con el fin de realizar un análisis detallado, en la Tabla 6.2 y la Figura 6.7 se aprecia el valor del índice Dice medio según la arquitectura de segmentación y el conjunto de datos. Dicha métrica ha sido

<i>Encoders</i>	<i>U-Net</i>		<i>U-Net++</i>		<i>DeepLabV3</i>	
Conjunto de datos	Base	Visual	Base	Visual	Base	Visual
ResNet18	0.8849	0.8894	0.8832	0.8959	0.8907	0.8895
ResNet34	0.8891	0.8907	0.8944	0.8907	0.8884	0.8888
ResNet50	0.8832	0.8911	0.8930	0.8927	0.8866	0.8858
EfficientNet-B1	0.8867	0.8884	0.8818	0.8874	0.8885	0.8893
EfficientNet-B2	0.8809	0.8805	0.8983	0.8921	0.8917	0.8884
EfficientNet-B3	0.8834	0.8799	0.8967	0.8880	0.8918	0.8916
EfficientNet-B4	0.8912	0.8908	0.8903	0.8871	0.8905	0.8974
EfficientNet-B5	0.8883	0.8932	0.8819	0.8888	0.8882	0.8810
EfficientNet-B6	0.8835	0.8856	0.8804	0.8905	0.8898	0.8920
DenseNet-121	0.8932	0.8924	0.8892	0.8927	-	-
DenseNet-161	0.8926	0.8893	0.8881	0.8920	-	-
DenseNet-201	0.8867	0.8969	0.8845	0.8969	-	-
DPN68	0.8756	0.8948	0.8941	0.8958	0.8899	0.8970
DPN98	0.8925	0.8923	0.8879	0.8915	0.8908	0.8935

Tabla 6.1: Coeficiente de Sørensen–Dice medio de cada experimento sobre el conjunto de test.

calculada promediando las métricas obtenidas para todos los *encoders* probados con cada arquitectura, según el conjunto de datos. Claramente, se aprecia como, para los 3 modelos de segmentación semántica, el rendimiento sobre el conjunto de test para el promedio de todos los *encoders* es superior cuando se utiliza como conjunto de datos aquel que incluye las *astronomic augmentations*. La variación más significativa se produce para la arquitectura *U-Net*, donde la mejoría en el coeficiente de Sørensen–Dice causada por la introducción de las *astronomic augmentations* es del 0.31 %. Por su parte, la mejoría producida para la arquitectura *U-Net ++* es del 0.28 % y la mejoría producida para la arquitectura *DeepLabV3* es del 0.06 %.

Cambiando de perspectiva el análisis realizado, sobre el conjunto de datos base la arquitectura que induce a unos mejores resultados es *DeepLabV3*. No obstante, la introducción de las *astronomic augmentations* provoca que el modelo con mejor rendimiento sobre el conjunto de test sea la red en *U* anidada, seguida de *DeepLabV3* y, en última posición, de la red en *U*. De estos resultados puede extraerse que el conjunto de datos visual provoca una mejoría que no tiene por qué ser proporcional en cada arquitectura de segmentación.

Una vez vistos los efectos positivos que provocan las *astronomic augmentations* realizando un agrupamiento de los resultados por arquitecturas de segmentación se procede a

Arquitectura	Dataset	Sørensen–Dice
U-Net	Base	0.886557
	Visual	0.889664
U-Net ++	Base	0.888843
	Visual	0.891643
DeepLabV3	Base	0.889718
	Visual	0.890391

Tabla 6.2: Coeficiente de Sørensen–Dice medio de cada arquitectura de segmentación según el conjunto de datos utilizado.

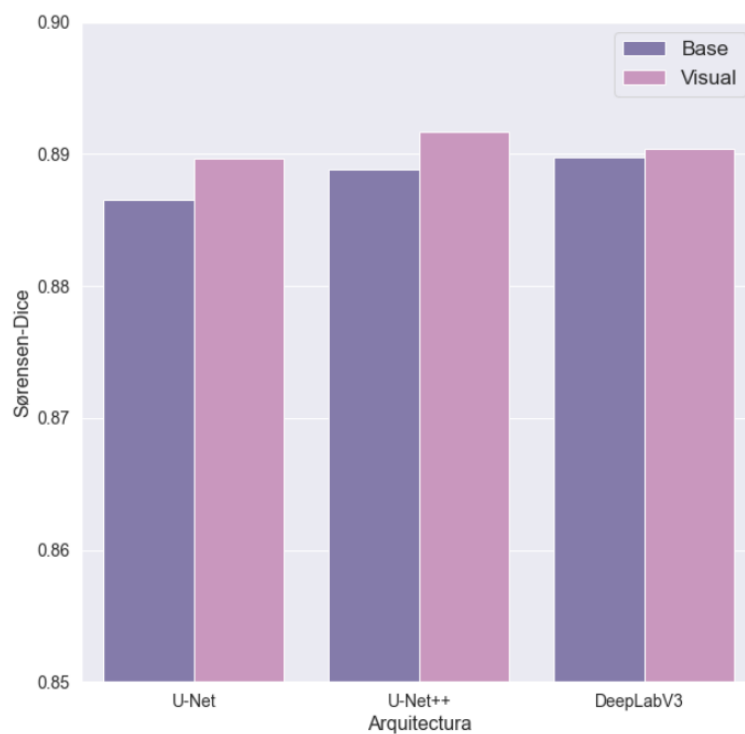


Figura 6.7: Gráfico de barras del coeficiente de Sørensen–Dice medio según arquitectura de segmentación y conjunto de datos.

realizar un estudio equivalente agrupando los resultados según el *encoder* utilizado. De esta manera, por cada *encoder* se calcula el promedio del coeficiente de Sørensen–Dice obtenido cuando se incrusta en las distintas arquitecturas de segmentación semántica, obteniendo los valores mostrados en la Tabla 6.3 y graficados en la Figura 6.8. De los 14 *encoders* utilizados, 11 obtienen mejores resultados cuando son entrenados con el conjunto de datos visual. Únicamente 3 de ellos, ResNet34, EfficientNetB2 y EfficientNetB3, consiguen un desempeño superior sobre el conjunto de test cuando no se utilizan las

Encoder	Base	Visual
ResNet18	0.886267	0.891600
ResNet34	0.890633	0.890067
ResNet50	0.887600	0.889867
EfficientNetB1	0.885667	0.888367
EfficientNetB2	0.890300	0.887000
EfficientNetB3	0.890633	0.886500
EfficientNetB4	0.890667	0.891767
EfficientNetB5	0.886133	0.887667
EfficientNetB6	0.884567	0.889367
DenseNet121	0.891200	0.892550
DenseNet161	0.890350	0.891100
DenseNet201	0.885600	0.896900
DPN68	0.886533	0.895867
DPN98	0.890400	0.892433

Tabla 6.3: Coeficiente de Sørensen–Dice medio de cada *encoder* según el conjunto de datos utilizado.

astronomic augmentations. Los 5 peores resultados, resaltados en rojo en la Tabla 6.3, son alcanzados mediante un proceso de entrenamiento alimentado por el conjunto de datos base, mientras que los 5 mejores resultados, marcados en color lima, son alcanzados a través de un proceso de aprendizaje de los algoritmos de aprendizaje profundo llevado a cabo mediante el conjunto de datos visual, es decir, el que lleva las *astronomic augmentations*. Especialmente interesante es el caso de la arquitectura DenseNet201 ya que forma parte de los 5 peores y los 5 mejores resultados cuando se usa el conjunto de datos base y visual respectivamente, siendo el aumento del coeficiente de Sørensen–Dice medio del 1.1%. Otro dato digno de mencionar es que los *encoders* ResNet34, EfficientNetB2 y EfficientnetB3, los únicos que no mejoran con la ampliación del conjunto de datos base, no se encuentran entre los mejores o peores *encoders*. Dicha característica puede reflejar que su capacidad de aprendizaje ya ha llegado al máximo únicamente utilizando las imágenes producto de la combinación de las bandas H, J e I, por lo que la introducción de las *astronomic augmentations* no provocará una mejoría del rendimiento sobre el conjunto de test.

Haciendo una valoración global de los resultados obtenidos, queda patente el efecto positivo que causa la introducción de las *astronomic augmentations* con el objetivo de determinar con precisión los límites de las galaxias del espacio distante. Las características físicas recogidas en cada combinación de bandas del espectro electromagnético son diferentes, características que son aprovechadas por la metodología de aprendizaje profundo

para realizar un segmentado de calidad. Tanto realizando un análisis centrado en las arquitecturas de segmentación como enfocado en los *encoders* se obtiene que el coeficiente de Sørensen–Dice medio del conjunto de test es mayor cuando se utiliza el conjunto de datos visual que cuando se utiliza su versión básica.



Figura 6.8: Gráfico de barras del coeficiente de Sørensen–Dice medio según *encoder* y conjunto de datos.

Para dotar de rigurosidad y significación estadística a los resultados obtenidos se efectuará un contraste de hipótesis con el objetivo de determinar si la introducción de las *astronomic augmentations* induce a un mejor resultado sobre el conjunto de test respecto al obtenido con el conjunto de datos base. Para ello, se plantea la realización de un test pareado de rangos con signo de Wilcoxon. Se trata de un test no paramétrico, esto es, no realiza suposiciones sobre la distribución subyacente de los datos, que constata si dos muestras proceden de poblaciones equidistribuidas. Es la alternativa no paramétrica al test de la *t*-Student, el cual realiza algunas suposiciones como la normalidad y homocedasticidad de los datos.

Para emplear la prueba pareada de rangos con signo de Wilcoxon es necesario que cada uno de los n sujetos, modelos de aprendizaje profundo en este caso, tenga 2 puntuaciones medidas, en este caso los índices Dice obtenidos para cada una de los 2 condiciones experimentales: su entrenamiento con el conjunto de datos base y su entrenamiento con el conjunto de datos visual. La hipótesis que se evalúa con la prueba pareada de rangos

con signo de Wilcoxon es si en las poblaciones subyacentes representadas por las muestras la diferencia de la mediana de las puntuaciones es o no igual a cero. Si se obtiene una diferencia significativa significa que hay una alta probabilidad de que las dos muestras o condiciones sean representativas de dos poblaciones diferentes. El test requiere las siguientes condiciones [28]:

1. Las diferentes puntuaciones extraídas de cada población han de ser independientes entre sí, lo cual se cumple.
2. Las puntuaciones se encuentran en un formato de ratio, lo cual se cumple.
3. La distribución de la resta de puntuaciones de las poblaciones representadas por las dos muestras es simétrica respecto a la mediana de la resta de las puntuaciones de las poblaciones. Típicamente no hay razones para sospechar de esta condición, tal y como ocurre en este caso.

De esta manera, se diseña un escenario donde se disponen de 39 sujetos, los 39 modelos que se han entrenado, en los que se han medido unas puntuaciones bajo 2 condiciones experimentales: su coeficiente de Sørensen–Dice medio sobre el conjunto de test cuando son entrenados sin y con las *astronomic augmentations* o, dicho de otra manera, con el conjunto de datos base y el visual. Se computa la versión de 2 colas del test, por ser esta más exigente que la versión unidireccional y disponer de un tamaño muestral razonable. Se fija un criterio de significación α del 5 %, y las hipótesis se definen tal y como se muestra a continuación

$$\mathcal{H}_0 : \theta_{\text{base}} = \theta_{\text{visual}}$$

$$\mathcal{H}_1 : \theta_{\text{base}} \neq \theta_{\text{visual}}$$

donde \mathcal{H}_0 es la hipótesis nula, \mathcal{H}_1 la hipótesis alternativa, θ_{base} el coeficiente de Sørensen–Dice mediano sobre el conjunto de test cuando el proceso de aprendizaje de los modelos *deep learning* es bajo el conjunto de datos base y θ_{visual} el coeficiente de Sørensen–Dice mediano sobre el conjunto de test cuando el proceso de aprendizaje de los modelos *deep learning* es bajo el conjunto de datos visual. Alternativamente, siendo $\theta_{\mathcal{D}} = \theta_{\text{base}} - \theta_{\text{visual}}$, la prueba de hipótesis se puede reformular de la siguiente manera:

$$\mathcal{H}_0 : \theta_{\mathcal{D}} = 0$$

$$\mathcal{H}_1 : \theta_{\mathcal{D}} \neq 0$$

Realizando el test de hipótesis, se obtiene un estadístico con valor 232.5 y un p -valor asociado de 0.0279. Si se realiza la variante de una cola:

$$\mathcal{H}_0 : \theta_{\mathcal{D}} < 0$$

$$\mathcal{H}_1 : \theta_{\mathcal{D}} > 0$$

se obtiene que el estadístico tiene un valor de 232.5 y un p -valor asociado de 0.014. Dichos resultados permiten rechazar ambas \mathcal{H}_0 con un nivel de significancia $\alpha = 5\%$ y asumir la diferencia de poblaciones de la que proceden las muestras. El test permite demostrar significativamente que el conjunto de datos visual es más beneficioso que el conjunto de datos base. Dicho de otra manera, la introducción de las *astronomic augmentations*, y su utilización simultánea para realizar inferencias de una misma galaxia, provoca una mejoría, estadísticamente probada, en el coeficiente de Sørensen–Dice obtenido sobre el conjunto de test.

6.4. Análisis visual del efecto de las convoluciones agrupadas

Una de las características más importantes del diseño experimental realizado es la introducción de las convoluciones en paralelo, o agrupadas, para procesar de manera independiente la información suministrada por las *astronomic augmentations*, en el caso de utilizar el conjunto de datos visual, de manera previa a su introducción a la red de segmentación correspondiente. Esta aproximación explota la información aportada por cada combinación de bandas del espectro electromagnético y no agrega las características de las distintas imágenes de manera prematura.

En las Figuras 6.9, 6.10 y 6.11 se aprecian los mapas de características obtenidos tras la aplicación de las convoluciones agrupadas para las galaxias DISKIRR-4532-cosmos, DISKSPH-14886-goodss y DISKSPH-18389-goodsn respectivamente. Los mapas de características han sido extraídos del modelo DeepLabV3 con *encoder* EfficientNetB4, la mejor configuración obtenida de todas las entrenadas a partir del conjunto de datos visual.

Las diferencias entre los mapas de características extraídos a partir de las distintas combinaciones de las bandas del espectro electromagnético son claras. En todos los casos, no existen dos mapas de características iguales, situación que explota la riqueza de cada imagen original y permite introducir características físicas más diferenciadas a partir de cada *input*. La concatenación de las 4 imágenes que se disponen para cada galaxia, en el caso de utilizar el conjunto de datos visual, para su introducción a la red de segmentación habría dificultado la extracción de tal riqueza visual independiente debido a la correlación que implícitamente existe entre los mapas de características de una convolución y la información en los distintos canales de los *inputs* de la misma.

Una vez aplicadas las convoluciones en paralelo, los 9 filtros resultantes de cada rama se concatenan en un único tensor con 36 canales. Dicho tensor es introducido a los modelos de segmentación para que agrupen las características extraídas para cada

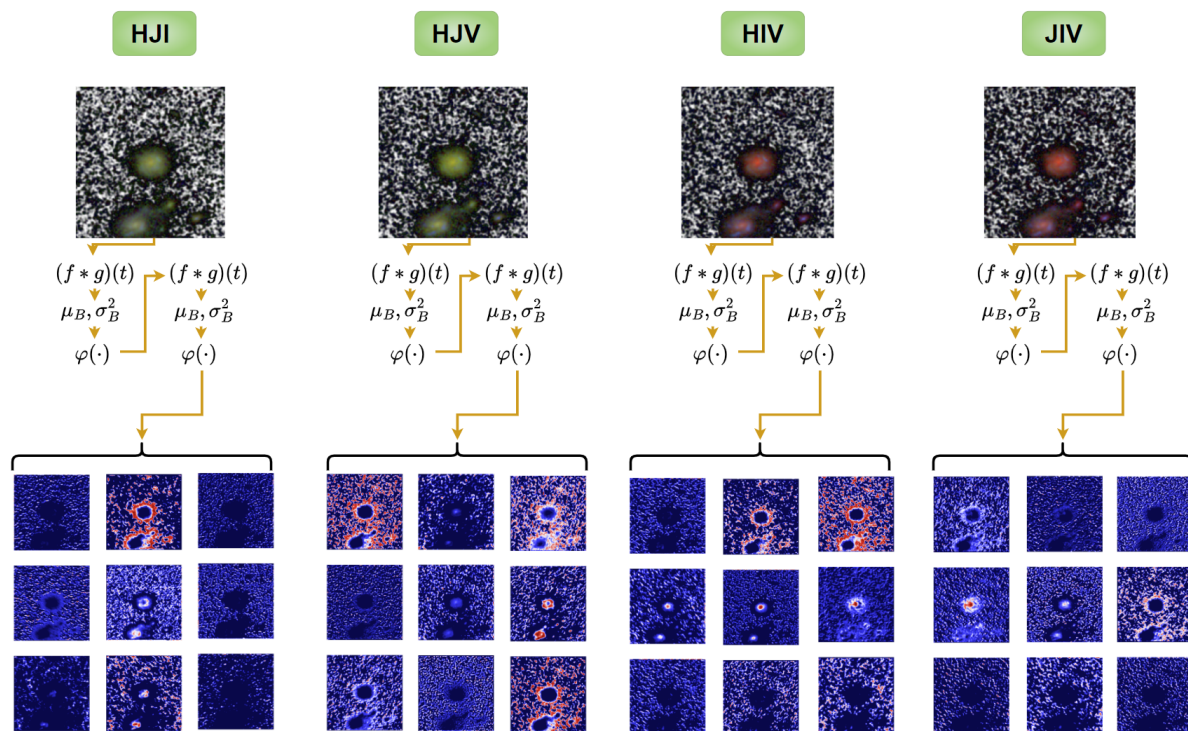


Figura 6.9: Mapas de características obtenidos tras la aplicación de las convoluciones agrupadas a la galaxia DISKIRR-4532-cosmos.

astronomic augmentation. Recordemos que dicha aproximación, formada por una etapa de procesamiento en paralelo, una etapa de concatenación y una etapa de procesamiento en cascada logra mejores resultados que la aproximación que únicamente utiliza el conjunto de datos base, por lo que los mapas de características computados por la fase compuesta por convoluciones agrupadas extraen detalles visuales de vital importancia que, posteriormente, son aprovechados de manera óptima por los modelos de segmentación semántica.

6.5. Análisis del mejor modelo

En esta sección se analizarán con un mayor grado de detalle los resultados obtenidos por el mejor modelo, entendiéndose como mejor modelo aquel que maximiza el coeficiente de Sørensen–Dice medio sobre la muestra de imágenes del conjunto de test. A pesar de ser palmaria la ventaja de introducir las *astronomic augmentations* para mejorar el proceso de aprendizaje de los modelos el algoritmo *deep learning* que mejores resultados ha obtenido es una red en U anidada, con un *encoder* basado en una EfficientNetB2, entrenada sobre el conjunto de datos base, por lo que dicho modelo será el analizado.

En la Tabla 6.4 se aprecia el valor del coeficiente de Sørensen–Dice, el *precision* y el *recall* medios sobre el conjunto de test, así como la prevalencia, agrupados por campos

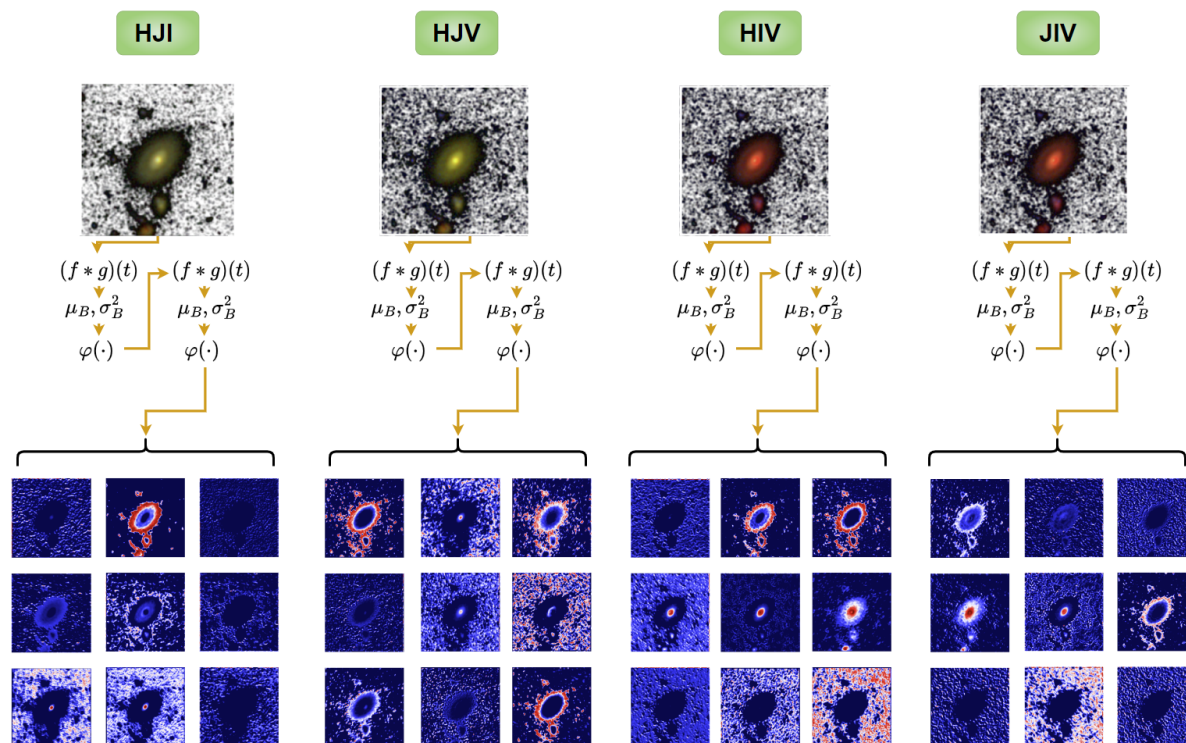


Figura 6.10: Mapas de características obtenidos tras la aplicación de las convoluciones agrupadas a la galaxia DISKSPH-14886-goodss.

de visión. Lo primero que salta a la vista es que, a pesar de que el índice Dice sobre el total de la muestra de test no llega a 0.90, para 4 de los 5 campos de exposición dicho umbral es superado. De esta manera, y atendiendo a la prevalencia de cada exposición en el conjunto de test, el coeficiente de Sørensen–Dice medio para las galaxias extraídas de los campos [GOODSN](#), [COSMOS](#), [UDS](#) y [EGS](#), las cuales representan el 85 % de la muestra de test, es superior al 90 %. Únicamente el índice Dice obtenido sobre las galaxias del campo de exposición [GOODSS](#) es inferior al 90 %, situándose en 0.8677. Dicho valor penaliza la media sobre la muestra total de test, a pesar de representar únicamente un 15 % de la misma. La utilización de un criterio más robusto que la media, como la mediana, para evaluar la calidad del segmentado sobre el conjunto de test evitaría dicha penalización excesiva. A pesar de ello, los resultados son excelentes, logrando una detección de bordes de gran calidad.

Dicha precisión en la tarea de asignar un tamaño a los cuerpos galácticos puede ser apreciada en la Figura 6.12, donde se muestran las 10 mejores segmentaciones realizadas por el mejor modelo. Por cada cuerpo, en la primera fila se muestra el borde real superpuesto a la galaxia, el borde predicho por la red superpuesto a la galaxia y ambos bordes superpuestos a la galaxia. La segunda fila muestra la máscara de segmentación real, la máscara de segmentación predicha y las diferencias entre ambas máscaras de segmentación.

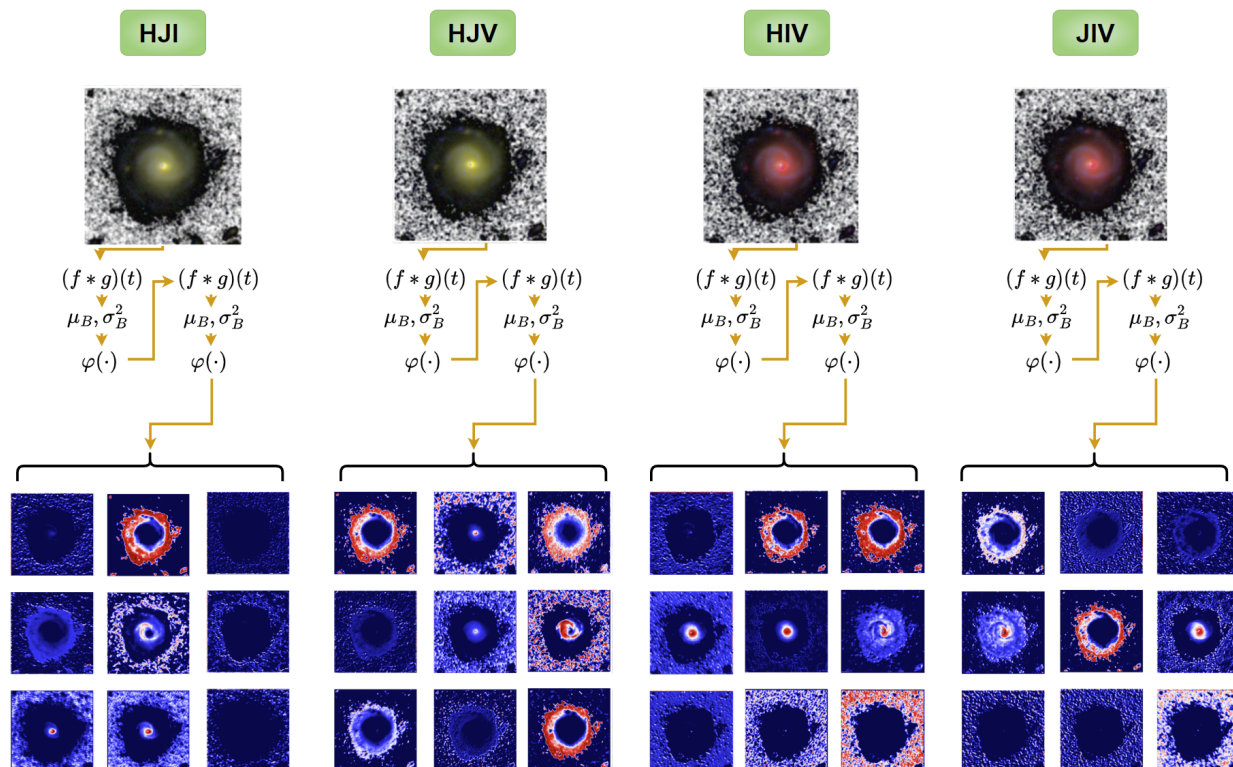


Figura 6.11: Mapas de características obtenidos tras la aplicación de las convoluciones agrupadas a la galaxia DISKSPH-18389-goodsn.

Métrica	GOODSS	GOODSN	COSMOS	UDS	EGS	Todos
<i>Precision</i>	0.8794	0.8869	0.8794	0.9097	0.9078	0.8914
<i>Recall</i>	0.8900	0.9400	0.9469	0.9250	0.9129	0.9268
Sørensen–Dice	0.8677	0.9036	0.9022	0.9101	0.9009	0.8983
Prevalencia	15	27	24	15	19	100

Tabla 6.4: *Precision*, *recall* y coeficiente de Sørensen–Dice medios sobre el conjunto de test según el campo de exposición y su prevalencia.

Dicha diferencia toma tonos verdosos cuando dichos píxeles están categorizados como galaxia en la máscara real y como ruido en la máscara predicha. Por otra parte, los tonos amarillentos indican presencia de píxeles categorizados como galaxia, de manera errónea, en la máscara predicha. Tonos negros indican total armonía entre predicción y realidad. La predominancia masiva del negro en dicha última representación para las 10 galaxias mostradas, intuida debido a la presencia de índices de Dice superiores al 95 %, muestra la excelencia alcanzada en la delimitación de los bordes. Apenas aparecen ligeras discrepancias en las zonas donde se encuentra la truncación de las galaxias.

Por otro lado, también es interesante estudiar los casos donde la segmentación de las galaxias es de peor calidad. Con ese fin, las 10 galaxias para las que se obtiene una delimitación de bordes menos precisa se encuentran en la Figura 6.13. Como es de esperar, los índices Dice son bastante más bajos que los anteriores, oscilando entre el 66 % y el 81 %. No obstante, observando cada caso de manera particular, los resultados quizás no sean tan malos como puedan parecer. Todas las inferencias guardan sentido, no se trata de predicciones cuyo patrón sea difícil de explicar. Son especialmente interesantes los casos de las galaxias DISK-20731-goodss, DISKSPH-21639-GOODSN y DISKSPH-944-cosmos, donde la red es capaz de aislar la galaxia de interés de otras galaxias que se encuentran alrededor, situación que no se contempla en la máscara real. Dichas inferencias, a pesar de obtener un coeficiente de Sørensen–Dice bajo, puede que reflejen de una manera más certera la truncación que la propia delimitación real. Para el resto de casos, o bien la truncación predicha es morfológicamente muy similar a la real pero más pequeña, situación en la que la métrica *precision* tiende a 1 y el *recall* cae, o bien la truncación predicha es morfológicamente muy similar a la real pero más amplia, situación en la que la métrica *precision* desciende y el *recall* tiende a la unidad.

En líneas generales, el comportamiento del mejor modelo es excelente. No sólo se alcanzan índices Dice medios superiores al 90 % para 4 de los 5 campos de exposición que conforman la muestra de test, si no que las inferencias de peor calidad parecen esconder un sentido físico más que razonable. De esta manera, dicho modelo puede ser utilizado para detectar discrepancias entre el segmentado realizado vía *deep learning* y el segmentado realizado manualmente por un experto, ya que este último puede contener errores al no tratarse de un proceso sencillo y claramente definido.

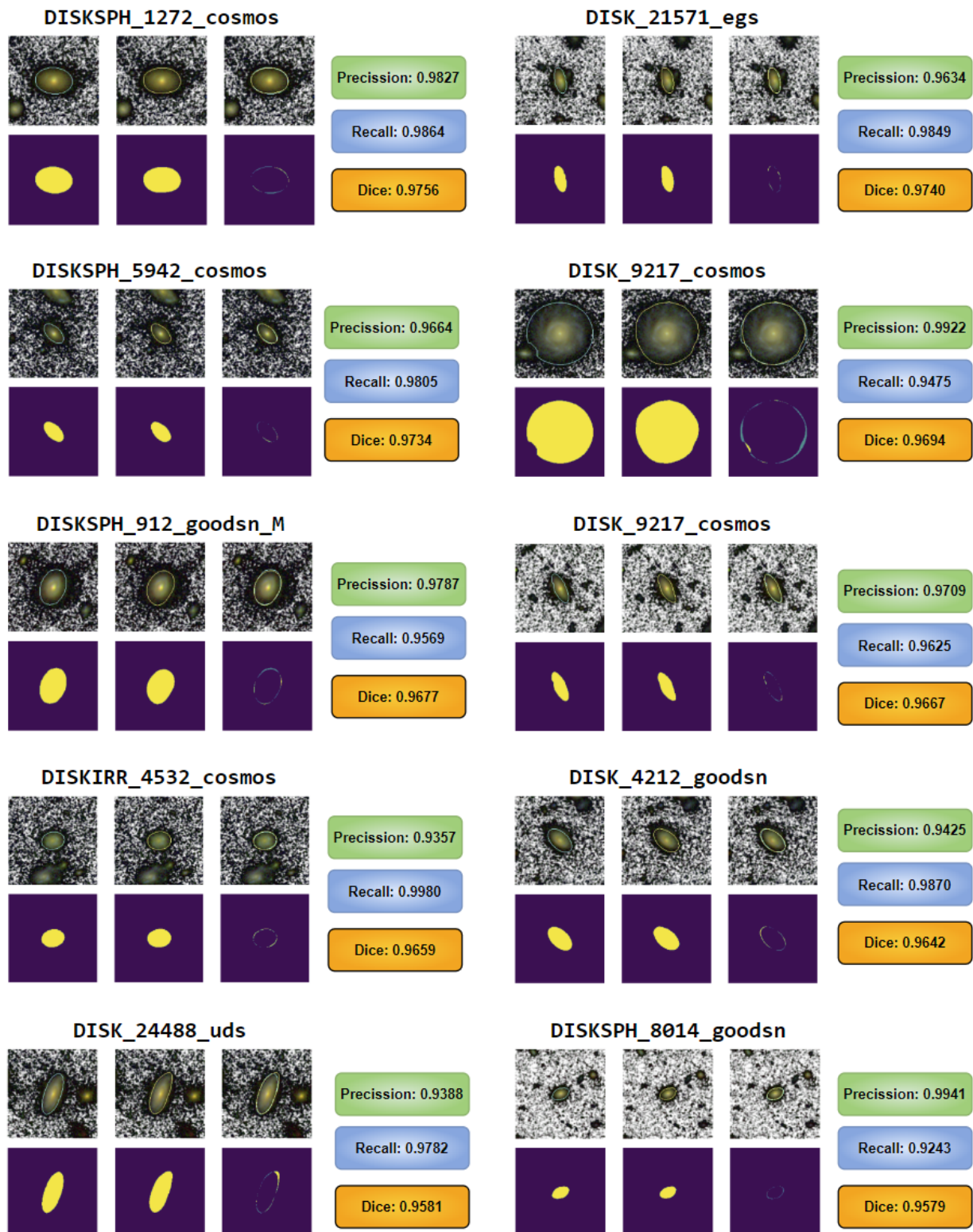


Figura 6.12: 10 mejores galaxias delimitadas por el mejor modelo. Para cada galaxia se tiene, de izquierda a derecha y de arriba a abajo, borde real, borde predicho, comparación de bordes, máscara real, máscara predicha y diferencia de máscaras.

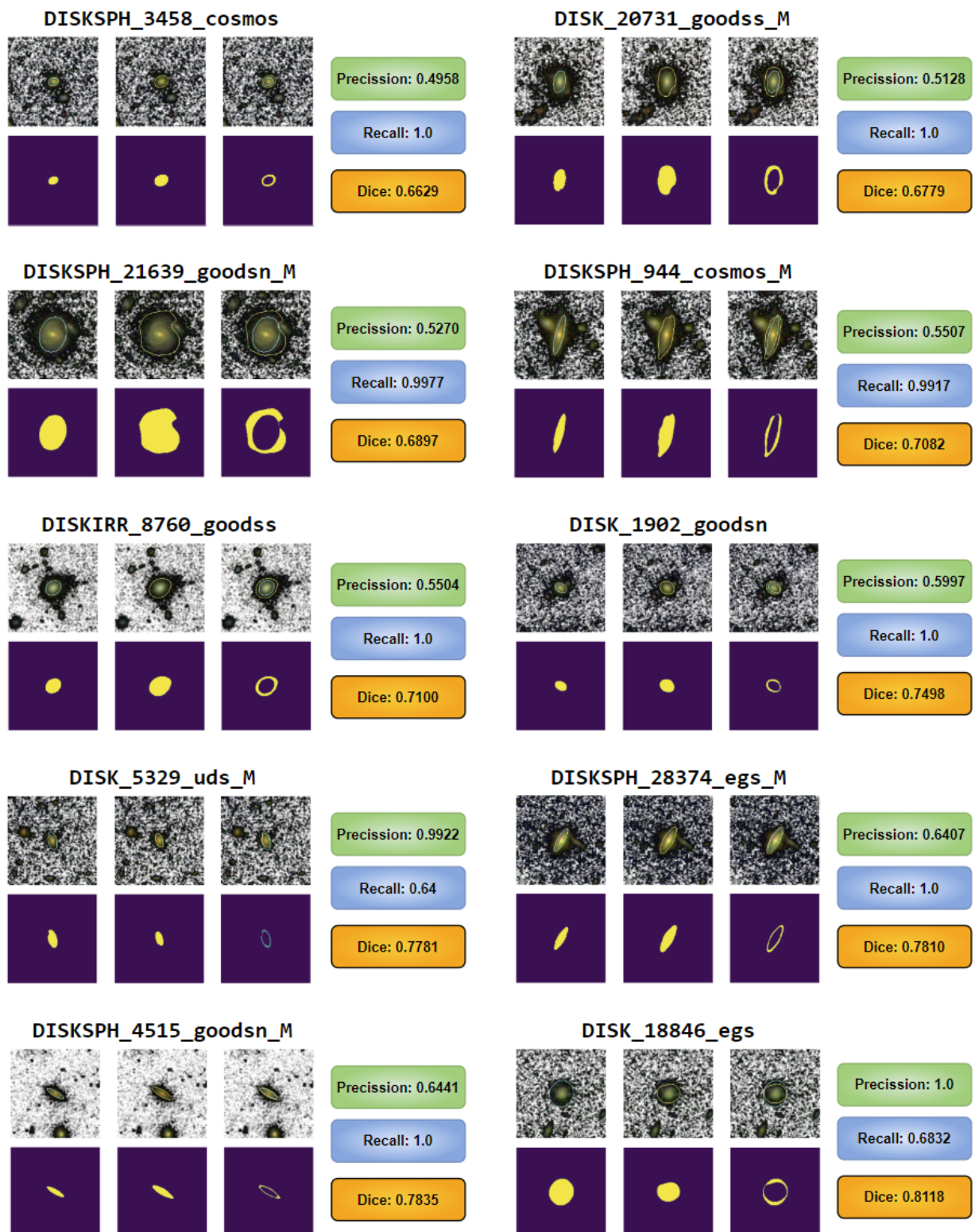


Figura 6.13: 10 peores galaxias delimitadas por el mejor modelo. Para cada galaxia se tiene, de izquierda a derecha y de arriba a abajo, borde real, borde predicho, comparación de bordes, máscara real, máscara predicha y diferencia de máscaras.

7: Conclusiones y líneas de trabajo futuras

Este capítulo está orientado ofrecer las conclusiones derivadas del desarrollo del presente documento así cómo posibles líneas futuras de investigación que amplíen y complementen los contenidos tratados.

7.1. Conclusiones

En este trabajo se ha logrado cumplir con creces el objetivo de desarrollar técnicas avanzadas de aprendizaje profundo que permitieran detectar bordes con una precisión razonable. La cota del 80 % en el coeficiente de Sørensen–Dice medio sobre el conjunto de test ha sido superada por un margen superior a un 10 %.

En ningún momento se han tenido problemas relevantes de infraajuste ni de sobreajuste. Esta situación viene motivada tanto por el desarrollo de algorítmica de aprendizaje profundo con una complejidad suficiente como por la inclusión de técnicas de regularización como el *data augmentation*, la adición de ruido, la penalización Ridge, la normalización de *batches* y la selección del mejor modelo sobre el conjunto de validación en el global de todo el entrenamiento.

En líneas generales, de las arquitecturas *deep learning* de segmentación semántica desarrolladas, aquella que mejores resultados ha obtenido es la *U-Net ++*, seguida de la arquitectura DeepLabV3 y, por último, la red en U clásica. Trasladando el análisis a los *encoders*, los que brindan un mayor rendimiento, en término general, pertenecen a las familias DenseNet, DPN y EfficientNet, en ese orden, siendo el rendimiento de las redes con bloques residuales, ResNet, algo inferior, aunque las diferencias son mínimas. El mejor modelo construido, una red en U anidada en la que las labores de codificar la información son realizadas por una EfficientNetB2, consigue un coeficiente de Sørensen–Dice medio

sobre el conjunto de test de 0.8983, un valor que roza el 90 % y representa una muy buena habilidad del modelo para estimar el tamaño de las galaxias del espacio distante.

Se ha demostrado estadísticamente, con un nivel de significancia del 5 % y mediante un test no paramétrico pareado de rangos con signo de Wilcoxon, como la utilización de las *astronomic augmentations*, variaciones en las combinaciones de bandas del espectro electromagnético con el fin de resaltar distintas zonas de las galaxias, induce a obtener mejores resultados sobre el conjunto de test que los obtenidos únicamente contando con las imágenes producto de la combinación de las bandas H, J e I, en ese orden.

De igual manera, se ha percibido visualmente la variedad de mapas de características obtenidos a partir de cada *astronomic augmentation* cuando estas son tratadas mediante convoluciones agrupadas de manera previa a su concatenación en la dimensión que representa el espectro electromagnético para su procesamiento por una arquitectura de segmentación semántica. Dicha aproximación permite evitar la generación de mapas de características correlados con varias *astronomic augmentations* desde las primeras capas de las redes y exprimir a fondo la información aportada por cada imagen antes de iniciar la combinación y reducción jerárquica de los *inputs* que es realizada por los *encoders*.

Por último, también se ha comprobado como aquellas instancias para las que las inferencias del mejor modelo son de peor calidad esconden una delimitación de bordes más que razonable para la vista humana, lo cual da pie a pensar que quizás los modelos de aprendizaje profundo puedan tener en cuenta características lumínicas y morfológicas que no son tenidas en cuenta por los expertos que realizan la asignación de tamaños de manera manual en base al estudio de los perfiles de brillo, color y masa.

7.2. Líneas de trabajo futuras

Se proponen las siguientes líneas de trabajo futuro en base a los resultados obtenidos en el presente proyecto:

- Generación de nuevas imágenes a partir de datos suministrados por el telescopio Hubble que puedan ser incluidas en los entrenamientos y potenciar el concepto de las *astronomic augmentations* desarrolladas en este trabajo.
- Estudiar el rendimiento de los modelos en función del campo de visión o de la morfología de los objetos galácticos y desarrollar un sistema que explote los puntos fuertes de cada modelo mediante una combinación inteligente de los mismos. Esto es, explorar la aplicación de aprendizaje por *ensembles* guiado por un análisis de datos detallado de las ventajas y desventajas de cada modelo.
- Revisar aquellos casos donde las inferencias de los mejores modelos difieran de manera significativa con los límites determinados mediante el estudio manual de los

perfiles de brillo, color y masa con el objetivo de, quizás, descubrir la razón de la divergencia de opiniones y explorar los pasos que pueden inducir a la concordancia de las mismas.

- Estudiar la adecuación de la aplicación de los modelos ajustados a conjuntos de datos astronómicos provenientes de otros estudios con el fin de determinar si las técnicas desarrolladas son robustas ante una modificación importante de los *inputs* o, en caso negativo, si la aplicación del *transfer learning* implica mejores resultados que el entrenamiento de modelos desde 0 en los nuevos conjuntos de datos.

Bibliografía

- [1] *A Guide To The Project Management Body Of Knowledge (PMBOK Guides) (6th ed.)*. Project Management Institute, 2017.
- [2] BUITRAGO, F., AND TRUJILLO, I. Revisiting the size evolution of galaxies: a new physically-motivated mass-size relation up to $z = 1$. *To be published* (2022).
- [3] CHAMAN, A., AND DOKMANIĆ, I. Truly shift-invariant convolutional neural networks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 3772–3782.
- [4] CHAMBA, N. Historical Perspectives on the Concept of Galaxy Size. *Research Notes of the American Astronomical Society* 4, 7 (2020), 117.
- [5] CHEN, L.-C., PAPANDREOU, G., KOKKINOS, I., MURPHY, K. P., AND YUILLE, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018), 834–848.
- [6] CHEN, L.-C., PAPANDREOU, G., SCHROFF, F., AND ADAM, H. Rethinking atrous convolution for semantic image segmentation. *ArXiv abs/1706.05587* (2017).
- [7] CHEN, Y., LI, J., XIAO, H., JIN, X., YAN, S., AND FENG, J. Dual path networks. *NeurIPS* (2017).
- [8] FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics* 36 (1980), 193–202.
- [9] FUKUSHIMA, K., AND MIYAKE, S. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Pattern Recognition* 15 (1982), 455–469.

- [10] GIUSTI, A., CIRESAN, D. C., MASCI, J., GAMBARDELLA, L. M., AND SCHMIDHUBER, J. Fast image scanning with deep max-pooling convolutional neural networks. *IEEE International Conference on Image Processing* (2013), 4034–4038.
- [11] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [12] GROGIN, N. A., KOCEVSKI, D. D., FABER, S. M., FERGUSON, H. C., KOEKE-MOER, A. M., RIESS, A. G., ACQUAVIVA, V., ALEXANDER, D. M., ALMAINI, O., ASHBY, M. L. N., BARDEN, M., BELL, E. F., BOURNAUD, F., BROWN, T. M., CAPUTI, K. I., CASERTANO, S., CASSATA, P., CASTELLANO, M., CHALLIS, P., CHARY, R.-R., CHEUNG, E., CIRASUOLO, M., CONSELICE, C. J., COORAY, A. R., CROTON, D. J., DADDI, E., DAHLEN, T., DAVÉ, R., DE MELLO, D. F., DEKEL, A., DICKINSON, M., DOLCH, T., DONLEY, J. L., DUNLOP, J. S., DUTTON, A. A., ELBAZ, D., FAZIO, G. G., FILIPPENKO, A. V., FINKELSTEIN, S. L., FONTANA, A., GARDNER, J. P., GARNAVICH, P. M., GAWISER, E., GIAVALISCO, M., GRAZIAN, A., GUO, Y., HATHI, N. P., HÄUSSLER, B., HOPKINS, P. F., HUANG, J.-S., HUANG, K.-H., JHA, S. W., KARTALTEPE, J. S., KIRSHNER, R. P., KOO, D. C., LAI, K., LEE, K.-S., LI, W., LOTZ, J. M., LUCAS, R. A., MADAU, P., MCCARTHY, P. J., MCGRATH, E. J., MCINTOSH, D. H., MCLURE, R. J., MOBASHER, B., MOUSTAKAS, L. A., MOZENA, M., NANDRA, K., NEWMAN, J. A., NIEMI, S.-M., NOESKE, K. G., PAPOVICH, C. J., PENTERICCI, L., POPE, A., PRIMACK, J. R., RAJAN, A., RAVINDRANATH, S., REDDY, N. A., RENZINI, A., RIX, H.-W., ROBAINA, A. R., RODNEY, S. A., ROSARIO, D. J., ROSATI, P., SALIMBENI, S., SCARLATA, C., SIANA, B., SIMARD, L., SMIDT, J., SOMERVILLE, R. S., SPINRAD, H., STRAUGHN, A. N., STROLGER, L.-G., TELFORD, O., TEPLITZ, H. I., TRUMP, J. R., VAN DER WEL, A., VILLFORTH, C., WECHSLER, R. H., WEINER, B. J., WIKLIND, T., WILD, V., WILSON, G., WUYTS, S., YAN, H.-J., AND YUN, M. S. Candels: The cosmic assembly near-infrared deep extragalactic legacy survey. *The Astrophysical Journal Supplement Series* 197, 2 (2011), 35.
- [13] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.
- [14] HICKS, M., AND FOSTER, J. Adapting scrum to managing a research group (2010).
- [15] HOERL, A. E., AND KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.
- [16] HOLSCHNEIDER, M., KRONLAND-MARTINET, R., MORLET, J., AND TCHAMITCHIAN, P. A real-time algorithm for signal analysis with the help of the wavelet transform. *Wavelets Springer Berlin Heidelberg* (1990), 286–297.

- [17] HUANG, G., LIU, Z., AND WEINBERGER, K. Q. Densely connected convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 2261–2269.
- [18] KAINZ, B. Deep learning: Equivariance and invariance. *Imperial College London*.
- [19] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25* (2012).
- [20] LECUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W., AND JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1* (1989), 541–551.
- [21] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86* (1998), 2278–2324.
- [22] MCCARTHY, J. What is artificial intelligence? *Stanford University* (2007).
- [23] MITCHELL, T. *Machine learning*. McGraw-Hill, 1997.
- [24] OTTER, D., MEDINA, J. R., AND KALITA, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems 32* (2021), 604–624.
- [25] PRASAD, T. K. Stars: Their properties. *Wright State University* (2014).
- [26] ROBERT, S., AND ARIAS, J. Review of risk management methods. *Business Intelligence Journal* (2011).
- [27] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation (2015). *MICCAI* (2015).
- [28] SHESKIN, D. J. *Handbook of Parametric and Nonparametric Statistical Procedures (5th ed.)*. Chapman and Hall/CRC, 2011.
- [29] SHORTEN, C., AND KHOSHGOFTAAR, T. A survey on image data augmentation for deep learning. *Journal of Big Data 6* (2019).
- [30] SU, H., MAJI, S., KALOGERAKIS, E., AND LEARNED-MILLER, E. G. Multi-view convolutional neural networks for 3d shape recognition (2015). *Proc. ICCV* (2015).
- [31] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S. E., ANGUELOV, D., ERHAN, D., VANHOUCHE, V., AND RABINOVICH, A. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 1–9.
- [32] TAN, M., AND LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv abs/1905.11946* (2019).

- [33] TRUJILLO, I., CHAMBA, N., AND KNAPEN, J. H. A physically motivated definition for the size of galaxies in an era of ultradeep imaging. *Monthly Notices of the Royal Astronomical Society* 493 (2020), 87–105.
- [34] XIE, S., GIRSHICK, R. B., DOLLÁR, P., TU, Z., AND HE, K. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5987–5995.
- [35] ZHANG, A., LIPTON, Z. C., LI, M., AND SMOLA, A. J. Dive into deep learning. *arXiv preprint arXiv:2106.11342* (2021).
- [36] ZHOU, Z., SIDDIQUEE, M. M. R., TAJBAKHS, N., AND LIANG, J. Unet++: A nested u-net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S... 11045* (2018), 3–11.