UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

CLASIFICACIÓN MORFOLÓGICA DE GALAXIAS A TRAVÉS DE APRENDIZAJE CONTRASTIVO AUTOSUPERVISADO

T E S I S

QUE PARA OPTAR POR EL GRADO DE

MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
PEDRO ARTURO FLORES SILVA

TUTOR
DR. GIBRÁN FUENTES PINEDA
IIMAS, UNAM

CIUDAD UNIVERSITARIA, CDMX. SEPTIEMBRE 2021

Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI will transform in the next several years.
-Adrew Ng

Most of human and animal learning is unsupervised learning. If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake. We know how to make the icing and the cherry, but we don't know how to make the cake. We need to solve the unsupervised learning problem before we can even think of getting to true AI.
-Yann LeCun

Agradecimientos

Académicos:

Al Dr. Gibrán Fuentes Pineda por darme su confianza, la paciencia, las enseñanzas, su tiempo, sus comentarios, por ser un excelente académico y una gran persona.

A la Dra. Wendy Aguilar, el Dr. Héctor Hernández, el Dr. Antonio Vázquez y el Dr. Boris Escalante, quienes fungieron como sinodales de este trabajo. Gracias por sus valiosos comentarios, sin ustedes este trabajo no sería lo que es hoy.

A la Universidad Nacional Autónoma de México, al Posgrado en Ciencia e Ingeniería de la Computación, a la Facultad de Ciencias y a la preparatoria Antonio Caso por brindarme una educación de calidad.

Al M. en C. e I. Efraín Condés por explicarme su trabajo, heredarme algunos códigos y por los regaños.

Al Consejo Nacional de Ciencia y Tecnología (Conacyt) por el apoyo económico brindado durante 4 semestres.

Personales:

A mis padres Rosendo Alejandro Flores Hernández y Rosa Elena Silva Montoya. Gracias a ustedes soy un hombre completo, responsable y feliz. Gracias por su amor, su felicidad, sus regaños, su esfuerzo, el apoyo que siempre me han brindado. Ustedes lo son todo para mi. Si pudiera elegir a mis padres, indudablemente los escogería. Los amo con todo mi corazón.

A Karen Rubí Jiménez López, mi pareja y amiga espacial. Gracias por escucharme, ir a aventuras juntos, ver monas chinas conmigo, jugar, por absolutamente todo. Ahora sé que eres una parte esencial de mi felicidad y de mi oscuridad. Te amo y te amaré siempre.

A la Dra. Laura Natalia Serkovic Loli, quien me enseño a ver por mis intereses e ideales. Gracias a lo aprendido juntos, hacer este trabajo no me fue tan difícil.

A la Fís. Judith Magdalena Vera López por sus enseñanzas, paciencia y por permitirme desarrollarme en la enseñanza de la física.

Al Dr. Darío Núñez Zúñiga. Gracias a su sabio consejo de titularme por tesis he llegado hasta aquí, un campo que quizá no esté muy relacionado con la física, pero que me ha hecho muy feliz. No olvidaré nunca sus palabras.

A entropía.ai y a su comunidad. Gracias por darme la oportunidad de crecer profesionalmente con ustedes. Sigamos por este apasionante camino de aprendizaje lleno de ilusiones y gente extraordinaria. Aplausos.

A mis modelos perrunos Mitzy, Vale y Tita que tanta felicidad han dado a mi vida.

A la vida/universo/Dios/partículas/causalidad o lo que sea que esté detrás del todo.

Índice general

Lista de símbolos

k Un escalar

$\vec{x}$ Un vector

$\vec{x}^\intercal$ Un vector transpuesto

$\sum_{i=1}^{N}$ Suma desde 1 hasta N

$\mathbb{R}$ Conjunto de número reales (única letra)

$\mathbb{W}$ Una matriz (excluye las letras R, I, Q, N)

w_{jk}^{ℓ} Peso de la k-ésima neurona de la capa $\ell - 1$ a la j-ésima neurona en la capa ℓ

b_j^{ℓ} Sesgo de la k-ésima neurona de la capa ℓ

$\frac{\partial \psi}{\partial \omega}$ Derivada parcial de ψ con respecto a la variable ω

$\frac{d \psi}{d \omega}$ Derivada total de ψ con respecto a la variable ω

$\vec{\nabla}\psi$ Gradiente de ψ

$\mathbb{A} \odot \mathbb{B}$ Producto de Hadamard entre las matrices $\mathbb{A}$ y $\mathbb{B}$

$\vec{x} \cdot \vec{y}$, $\langle \vec{x} | \vec{y} \rangle$ Producto escalar o producto punto entre los vectores $\vec{x}$ e $\vec{y}$

$\int_b^a \psi(x) dx$ Integral definida de a a b de la función ψ

$(\psi \star \phi)(t)$ Operación de convolución entre las funciones o señales ψ y ϕ

$||\vec{x}||_2$ Norma euclídea del vector $\vec{x}$

Glosario

ETGs Galaxias de tipo temprano (*Early Type Galaxies*)

LTGs Galaxias de tipo tardío (*Late Type Galaxies*)

SB LTGs barradas

SA LTGs no barradas

E Galaxias elípticas: $\{cE, E, E^{+}\}$

S0 Galaxias lenticulares: $\{S0^{-}, S0^{o}, S0^{+}, S0a\}$

S Galaxias espirales: $\{Sa, Sab, Sb, Sbc, Sc, Scd, Sd, Sdm, Sm\}$

I Galaxias irregulares: $\{Im\}$

SDSS *Sloan Digital Sky Survey*

DES *Dark Energy Survey*

LSST *Large Synoptic Survey Telescope*

NSA *Nasa-Sloan Atlas*

CNN Red neuronal convolucional (*Convolutional Neural Network*)

SVM Máquinas de soporte vectorial (*Support-Vector Machine*)

KNN k vecinos más cercanos (k*-Nearest Neighbors*)

LinearReg Regresión lineal

Locally Weighted Reg Regresión local

RMS Valor cuadrático medio (*Root mean square*)

MLP Perceptrón multicapa (*Multilayer perceptron*)

ResNet Redes neuronales residuales (*Residual neural network*)

ReLU *Rectified Linear Unit*

VQ-VAE *Vector Quantized Variational Autoencoders*

HC Agrupamiento jerárquico (*Hierarchical clustering*)

SHAP *SHapley Additive exPlanations*

SimCLR *A Simple Framework for Contrastive Learning of Visual Representations*

Clasificación morfológica de galaxias a través de aprendizaje contrastivo autosupervisado

por

Pedro Arturo Flores Silva

Resumen

Este trabajo aborda uno de los problemas abiertos de la ciencia astronómica, asociado, principalmente, al rápido avance tecnológico que se ha dado en recientes años. Este problema se refiere a la clasificación morfológica de galaxias, la cual cuenta con una gran cantidad de datos que, incluso, no han sido explorados. Debido a la naturaleza del universo, es altamente probable que la distribución de clases de galaxias dentro de la gran mayoría de los estudios del cielo (*sky surveys*) no sea uniforme, siendo las clases más desbalanceadas aquellas galaxias cuya constitución morfológica sea irregular. Más aún, la existencia de diversos esquemas de clasificación deja en claro que el problema de la clasificación morfológica de galaxias es sumamente complejo, incluso, para los expertos astrónomos.

Aún con estas cuestiones, la tarea ha sido ampliamente estudiada desde una perspectiva computacional, en particular, desde el aprendizaje de máquinas y profundo, obteniéndose resultados excepcionalmente buenos en la distinción entre las clases tempranas y tardías. Sin embargo, al extender la cantidad de clases, el rendimiento de los modelos se ve significativamente reducido. Además, la mayoría de los trabajos relacionados emplean una perspectiva completamente supervisada, la cual, debido a la limitación de etiquetas disponibles y la calidad de las imágenes, podría sesgar los resultados obtenidos. Así mismo, en dichos trabajos parece evitarse y/o ignorarse una característica fundamental de los datos de esta índole: el desbalance entre clase. Esto se ve reflejado en el fuerte interés de la métrica de evaluación de exactitud, sin embargo, ésta es un mal indicador del rendimiento de un sistema que se enfrenta a conjuntos de datos inherentemente desbalanceados.

Al considerar las características comunes de la gran mayoría de estudios, este trabajo se centró en estudiar el problema de la clasificación morfológica de galaxias a través de los métodos del estado del arte auto- y semi- supervisados mediante el uso de un conjunto de datos no etiquetado, cuya calidad o resolución puede ser considerada no homogénea, y un conjunto etiquetado, considerando cinco clases, desbalanceado y de calidad homogénea.

Los experimentos realizados pueden ser sintetizados en:

I) Estudio del impacto en el rendimiento de una red profunda, al emplear un tratamiento de datos que considere las características de las galaxias, con el fin de mejorar el rendimiento de aquellas clases más desbalanceadas.

II) Estudio del rendimiento de una red profunda, que junto al algoritmo no supervisado de K-medias, actualiza y pseudobalancea las clases inferidas por cada época, con el fin de aumentar la tasa de verdaderos positivos en las clases más desbalanceadas.

III) Demostración de la efectividad de los métodos empleados.

IV) Comparación de los resultados obtenidos entre tres redes con distintos codificadores de características.

V) Implementación de una serie de algoritmos, que permitiesen visualizar: a) regiones de decisión para clasificación, b) distribuciones estimadas y reales de clases en los conjuntos etiquetados y no etiquetados, c) cuadrículas de activación e inversión de características de una cantidad determinada de imágenes, con el fin de comprender, visualmente, los pesos aprendidos por cada capa de las redes empleadas.

De los resultados cuantitativos más destacables son los obtenidos por la red semisupervisada entrenada con un conjunto de transformaciones especial y desbalance de clases, con valores de exhaustividad para las clases:

{0: cE, E, E^+, $S0^-$, 1: $S0^o$, $S0^+$, $S0a$, 2: Sa, Sab, Sb, 3: Sbc, Sc, Scd y 4: Sd, Sdm, Sm, Im} de $90.8\,\%$, $55.7\,\%$, $70.9\,\%$, $78.5\,\%$ y $51.2\,\%$ respectivamente, con una exactitud de $75.5\,\%$. En cuanto los valores de precisión se obtuvieron, respectivamente, los valores $86.8\,\%$, $62.3\,\%$, $71.3\,\%$, $75.3\,\%$ y $67.7\,\%$.
Para este caso, las proporciones relativas con respecto a la clase mayoritaria (clase dos) fueron de 0.93, 0.53, 0.79 para 0, 1, 3 y 4 respectivamente.
Para el caso balanceado se obtuvieron valores de exhaustividad para las clases 0, 1, 2, 3 y 4 de $89.9\,\%$, $63.4\,\%$, $75.4\,\%$, $52.8\,\%$ y $91.5\,\%$, con una exactitud de $72.2\,\%$ y valores de precisión correspondientes a $87.9\,\%$, $59.0\,\%$, $68.6\,\%$, $81.9\,\%$ y $30.2\,\%$
Por otro lado, se comprobó, por separado, que los puntos I) y II) sí mejoran la proporción de verdaderos positivos para las clases más desbalanceadas.

En cuanto a los resultados cualitativos, se encontró que las regiones de decisión para cada clase presentan grupos de patrones particulares, similares a aquellos buscados por los expertos astrónomos al momento de clasificar. Adicionalmente, se encontró que las distribuciones de clases estimada y real preservan una gran similitud visual, con lo cual fue posible estimar la distribución de clases en el conjunto no etiquetado.

El código generado durante esta investigación puede encontrarse en https://github.com/Pedri0/Modified-SimCLR.

Capítulo 1

Introducción

1.1. De la nada hasta hoy: Breve repaso histórico de la astronomía

La astronomía es una ciencia con una historia muy particular, no se sabe con certeza cuál civilización fue la primera en observar, de manera metodológica, los astros. Sin embargo, se sabe que una cantidad importante de éstas centró su atención y esfuerzos en aquellos cuerpos misteriosos con el fin de explicar el origen del universo. La formalización de esta ciencia, o lo que se conoce como la fundación de la astronomía moderna, se dió en el Renacimiento (siglo XVI) gracias al modelo matemático heliocéntrico del movimiento planetario propuesto por Copérnico. Las contribuciones realizadas por Copérnico son consideradas como la base de la revolución científica renacentista, ya que inspiró a los ilustres de su época a sentar y desarrollar las bases de la ciencia y astronomía modernas. Entre los principales contribuyentes de esta época encontramos a Tycho Brahe, considerado uno de los grandes observadores del cielo antes de la invención del telescopio y quien logró medir las posiciones de las estrellas y planetas de una forma muy precisa y sistemática para la época. Johannes Kepler, quien colaboró con Brahe y postuló las tres leyes del movimiento planetario y Galileó Galilei, cuyo trabajo más destacado fue el perfeccionamiento del telescopio con el cual logró la observación y estudio de los astros como la luna, las fases de Venus, las lunas de Júpiter, las manchas solares, entre otros.

A partir del siglo XVII, la ciencia en general se desarrolló de una manera acelerada. En particular, en el campo de la astronomía, las observaciones se intensificaron, los telescopios, así como las mediciones se perfeccionaron [1]. Como resultado de esto surgen nuevas teorías, nuevos fenómenos que requirieron ser explicados, así como el descubrimiento de diferentes cuerpos celestes.

Gracias a los grandes avances de la física durante los siglos XIX y XX, nació y se desarrolló la astrofísica, la cual estudia la composición, estructura y evolución de las estrellas, los planetas, las galaxias, la radiación de fondo de microondas, entre otros. Así mismo, durante este periodo de tiempo, las investigaciones en astronomía se centraron en el perfeccionamiento de los instrumentos de observación, se construyeron telescopios de grandes dimensiones, se lograron avances significativos en el área de la óptica instrumental y se creó un método para la medición de distancias estelares. Dos avances tecnológicos de especial mención, desarrollados durante este período, son el nacimiento de la fotometría moderna y la obtención de las primeras fotografías astronómicas en diferentes longitudes de onda, siendo ésta última una gran impulsora del desarrollo de la astronomía en general.

A la fecha, la astronomía puede dividirse en dos grandes ramas: teórica y observacional, donde la primera de ellas consiste, a grandes rasgos, en la creación y verificación de modelos basados en observaciones o simulaciones, mientras que la segunda consiste en el estudio de la estructura, evolución y origen del universo a partir de las propiedades físicas y químicas a través de observaciones derivadas de telescopios y/o detectores.

1.2. Clasificación morfológica de galaxias

El interés de este trabajo se centra en el área de la astronomía observacional, en particular en la clasificación morfológica de galaxias, sin embargo, este problema será estudiado desde la perspectiva del aprendizaje profundo computacional. A pesar de ello, debe entenderse, cuando menos, las nociones generales de este tópico astronómico. Es por esta razón por la cual esta sección detalla algunos conceptos inherentes a la clasificación morfológica de galaxias.

El pionero de la clasificación morfológica de galaxias fue Edwin Hubble, quien en 1926 publicó su trabajo titulado Extragalactic Nebulae [2], en el cual se clasifican las galaxias cercanas dentro de dos tipos principales: Galaxias de tipo Temprano (ETGs por las siglas en inglés de *Early Type Galaxies*) y Galaxias de tipo Tardío (LTGs por las siglas en inglés de *Late Type Galaxies*). Dentro de las ETGs se encuentran galaxias elípticas y galaxias lenticulares, cuyo componente característico es la posesión de un bulbo dominante. Este tipo de galaxias poseen, en su mayoría, población estelar antigua y además son masivas. En cambio, las galaxias de tipo LTGs poseen prominentes brazos espirales o no presentan una morfología bien definida (galaxias irregulares), su población estelar es mucho más joven y son consideradas sistemas altamente fértiles en cuanto a la formación de estrellas.

Estudios posteriores refinaron el método de clasificación de Hubble al dividir la clase LTGs en dos subclases: Galaxias barradas (SB) y no barradas (SA), que a su vez pueden ser divididas nuevamente al considerar la forma de sus brazos espirales [3].

De acuerdo a [3], al localizar las ETGs es posible mapear a gran escala la estructura del universo, ya que este tipo de galaxias están situadas en los centros de grupos de galaxias, poseen menos gas y velocidades de dispersión mucho mayores comparadas con las LTGs. Es por ello que la morfología de las galaxias, así como su clasificación juegan un papel sumamente importante para la astronomía extragaláctica.

El acelerado desarrollo tecnológico en diversas áreas científicas ha permitido un progreso excepcional en la astronomía observacional, pues la construcción de grandes observatorios astronómicos, así como la construcción de nuevos telescopios e instrumentos a bordo de satélites y robots exploradores permiten la obtención de datos fotométricos y exploración de miles y millones de galaxias en diversas zonas del espacio. A pesar de que, en principio, es posible obtener los datos fotométricos de forma aleatoria a lo largo del universo observable, esto no se hace así. Una forma más coherente y ordenada de obtener dichos datos consiste en centrar la atención en determinadas regiones y/o cuerpos de interés. A esta metodología se le conoce como exploraciones astronómicas (*sky surveys*), las cuales pueden ser identificadas al considerar la motivación científica, la estrategia de adquisición de datos, el régimen de longitud de onda, el tipo de observaciones, las regiones de interés, así como la profundidad, el carácter temporal (una contra varias épocas) y la elección de cobertura individual. Es decir, por un lado, puede elegirse cubrir un área del espacio con todos los objetos posibles (vista panorámica), mientras que por otro lado, puede optarse por observar una lista definida de objetos de interés [4].

Desde la década de los noventas, las exploraciones astronómicas se han convertido en la principal fuente de datos en astronomía y se cree que la cantidad de datos obtenida por dichos estudios se duplica al ritmo de la ley de Moore, cada año o año y medio [4]. Este hecho nos revela uno de los grandes problemas existentes en la astronomía observacional; si bien es cierto que, los datos obtenidos tienen un gran valor científico, una cantidad importante de los mismos no han sido analizados por expertos en el área debido a dos principales razones; la primera de ellas ya ha sido mencionada, es prácticamente imposible analizar la gran cantidad de datos existentes de manera tradicional y, además, el análisis de éstos es una tarea compleja y complicada de realizar que requiere de gran experticia en el campo y mucho tiempo.

En un intento de superar el problema del análisis de datos astronómicos, así como extender los catálogos de galaxias clasificadas, el proyecto *Galaxy Zoo* permite a personas aficionadas clasificar un conjunto de datos fotométricos al contestar una serie de preguntas. En la primera fase del proyecto se obtuvieron más de 4×10^7

etiquetas individuales realizadas por aproximadamente 10^5 participantes, considerando únicamente las clases espiral y elíptica [3, 5]. A causa de la gran respuesta obtenida durante la primera fase, se planteó una segunda fase más ambiciosa, en la cual se consideraron características más detalladas para la clasificación, tales como la forma de los brazos espirales, la existencia de barras, forma de los bulbos, entre muchos otros. Para esta segunda fase, de tres años de duración, se obtuvieron etiquetas para 304 mil galaxias presentes en el *Sloan Digital Sky Survey* (SDSS)[1] [7, 8]. A pesar del gran éxito de *Galaxy Zoo*, el problema de analizar una gran cantidad de datos sigue presente. Al considerar la cantidad de tiempo requerido en las dos fases del proyecto, se estima que los conjuntos de datos adquiridos por los proyectos *Dark Energy Survey* (DES) y *Large Synoptic Survey Telescope* (LSST) serían clasificados en más de cien años, si éstos ingresaran al proyecto de *Galaxy Zoo* [8]. Por otro lado, debe considerarse que la gran mayoría de los proyectos se encuentran en continuo desarrollo. Es por estas razones que se requiere un método de análisis eficiente, rápido y con la menor cantidad de sesgo posible. Debido a las características del problema en sí, así como la necesidad de un método que cumpla las características antes mencionadas, se han propuesto a las técnicas del aprendizaje de máquina y aprendizaje profundo como las opciones ideales para atacar este problema [3, 4, 8].

1.3. Aprendizaje profundo y de máquina en la clasificación morfológica de galaxias

La primera aplicación del aprendizaje de máquina para la clasificación de galaxias data del año 1992 por S. Lombardi y colaboradores, quienes entrenaron una red neuronal densa de trece neuronas. El método de entrenamiento de esta red consideró algunas propiedades físicas de las galaxias y fue entrenada durante 1.5 millones de épocas para la clasificación de cinco tipos de galaxias, obteniéndose una exactitud del 64 % en el conjunto de prueba [9]. Pese a que los resultados obtenidos por este estudio fueron muy prometedores y la importancia de la clasificación morfológica de galaxias, conforme con [8] la cantidad de estudios que utilizan las técnicas de aprendizaje de máquina y/o profundo para este fin son relativamente pequeñas. En cambio, la solución de este problema desde esta perspectiva computacional enfrenta, también, grandes retos; uno de ellos consiste en responder a las preguntas: ¿cómo realizar esta clasificación a través de características no paramétricas o propiedades físicas medibles?, ¿cuáles son las características paramétricas más útiles? y/o ¿cuáles son los mejores métodos para enfrentar esta tarea? [3]. Otro de los grandes retos a enfrentar, se encuentra sumamente ligado a la cantidad de etiquetas existentes en los diversos conjuntos de datos; es bien sabido que, tanto el aprendizaje profundo, como el aprendizaje de máquina requieren de una cantidad considerable de datos para el correcto aprendizaje y generalización de las características inherentes a los datos y debido a la relativa falta de etiquetas, es necesario un método que no dependa fuertemente de las mismas, tal y como lo hace el aprendizaje supervisado.

Estudios enfocados en la solución de este problema a través del aprendizaje de maquina y/o profundo emplean, en su mayoría, la perspectiva supervisada, limitando así el aprendizaje de las representaciones al restringir los conjuntos de entrenamiento. A pesar de ello, dichos estudios se concentran en responder las cuestiones anteriormente planteadas. A modo de resumen ilustrativo, la tabla 1-1 muestra los mejores resultados, la técnica empleada y las características de clasificación de algunos de estos estudios.

[1] SDSS es uno de los conjuntos de datos más empleados para el estudio astronómico en el cual se han estado adquiriendo datos fotométricos desde 1998 [3]. Los datos publicados por este proyecto en la séptima liberación de datos del año 2009, contiene datos fotométricos de 357 millones de objetos únicos [6].

Método	Características			Exactitud
	# imágenes	# clases	¿Desbalance?	
CNN [3]	104787	2	Sí	99.1 %
CNN [8]	11381	2	No	92.4 %
CNN [8]	12381	2	Sí	90.6 %
CNN [3]	138430	3	Sí	81.8 %
CNN [3]	138430	7	Sí	77.6 %
SVM [3]	138430	7	Sí	62.6 %
SVM [3]	138430	3	Sí	74.6 %
KNN [8]	11381	2	No	78.2 %
LinearReg [8]	12381	2	No	68.2 %
MLP [10]	310	5	Sí	50.3 %
MLP [10]	310	3	Sí	90.6 %
Locally Weighted Reg [10]	310	5	Sí	49.9 %
Locally Weighted Reg [10]	310	3	Sí	91.8 %
MLP [11]	830	16	Sí	1.8 RMS

Tabla 1-1: Descripción general de estudios relacionados con la clasificación morfológica de galaxias. *CNN*: redes neuronales convolucionales, *SVM*: maquinas de soporte vectorial, *KNN*: K vecinos más cercanos, *LinearReg*: regresión lineal, *MLP*: Perceptrón multicapa, *Locally Weighted Reg*: regresión local. Las principales fuentes de información fueron: arXiv, Mendeley y Google Scholar.

De la tabla anterior (1-1) podemos destacar que; el problema de clasificación reducido a dos clases (tempranas y tardías) es el más ampliamente estudiado y los resultados obtenidos por [3, 8] son buenos (superan el 90 % de exactitud). Más aún, los autores atribuyen los errores de clasificación a un etiquetado incorrecto en Galaxy Zoo. En contraste, la consideración de más de dos clases presenta un gran reto, pues los resultados obtenidos por los diversos autores muestran una tendencia decreciente con respecto al rendimiento de sus modelos al considerar cada vez más clases. Este hecho puede deberse a dos principales razones: el desbalance de clases y la calidad de los datos. Por un lado, el desbalance de clases, inherente a la clasificación de galaxias [3], representa una enorme dificultad para los algoritmos del aprendizaje automático, ya que éstos pueden no aprender correctamente las representaciones de las clases minoritarias y/o puede resultar en un sobreajuste en aquellas clases dominantes. Para superar esta barrera, es común aplicar las técnicas de submuestreo y sobremuestreo de datos, con el fin de balancear las clases. Sin embargo, la técnica de sobremuestreo implica la generación de datos sintéticos que no necesariamente son útiles, mientras que la técnica de submuestreo implica una reducción de las clases dominantes provocando así una reducción de los datos disponibles. Por otro lado, la calidad de datos empleados es también un parámetro sumamente importante; dado que la clasificación morfológica de galaxias depende de datos fotométricos, idealmente se espera que éstos presenten únicamente la información de interés, la menor cantidad de ruido y además ser de alta calidad. No obstante, estas condiciones son extremadamente difíciles de conseguir. Pues los objetos de estudio se encuentran a distancias muy lejanas, la vista o perspectiva de los objetos está limitada debido a la posición relativa entre los mismos y los instrumentos, las limitaciones y errores al momento de la adquisición debido al instrumental, la dinámica interestelar no es estacionaría, por lo que los objetos pueden sufrir cambios a lo largo del tiempo. Además, dentro de esta dinámica se presentan tres fenómenos físicos que pueden afectar la calidad de los datos; estos tres fenómenos están relacionados con el cambio de la longitud de onda de la radiación electromagnética que capturan los detectores, siendo cada uno de distinta naturaleza.

1.4. Objetivos

Una vez conocidas la relevancia y dificultades que presenta la tarea de la clasificación morfológica de galaxias, esta investigación se centró en explorar algunos métodos que reduzcan el impacto sobre la tarea debido al desbalance de clases. También, para superar la barrera de la dependencia de etiquetas, se aplicó un esquema semisupervisado, el cual permite un aprendizaje de representaciones generales sin la necesidad de etiquetas,

para posteriormente, ajustar dichas representaciones al utilizar una cantidad determinada de etiquetas.

Para estos dos fines se plantearon una serie de objetivos específicos:

- Implementar la arquitectura del estado del arte del aprendizaje semisupervisado, conocida como SimCLR [12], para la tarea de clasificación morfológica de cinco clases de galaxias.
- Estudiar, analizar y comprender, experimentalmente, el impacto que tiene el preprocesamiento de los datos en el aprendizaje de las representaciones generales.
- A partir del mejor conjunto de preprocesamiento, realizar el aprendizaje de las representaciones generales a través de la arquitectura del estado de arte, con el fin de realizar un análisis comparativo, entre este nuevo preprocesamiento y el preprocesamiento propuesto por dicha arquitectura.
- Evaluar las técnicas de transferencia de conocimiento y ajuste fino al utilizar aproximadamente 2 % de las etiquetas y las representaciones generales previamente aprendidas.
- Explorar una posible solución para el problema del desbalance de clases a través de un método de exploración-explotación como lo es el algoritmo de K-medias.
- Realizar una exploración cualitativa y visual de algunas características aprendidas por los modelos implementados.

1.5. Hipótesis

Una de las principales motivaciones de este trabajo radica en el estudio de la clasificación morfológica de galaxias a través del aprendizaje profundo. A pesar de que este tópico ha sido estudiado en una amplia variedad de investigaciones, se ha observado una tendencia clara hacía la aplicación de métodos totalmente supervisados. Sin embargo, dado el crecimiento exponencial de datos y la imposibilidad de etiquetarlos en su totalidad, este trabajo parte de la hipótesis de que al añadir información proveniente de un volumen considerable de imágenes no etiquetadas será posible brindar una alternativa robusta, plausible, efectiva y rápida que permita la exploración y extracción automatizada de representaciones morfológicas relevantes en grandes conjuntos de datos. En particular, se explorarán métodos de aprendizaje autosupervisado constrastivo para el aprendizaje de representaciones relevantes a partir de un conjunto de imágenes no etiquetadas.

En esta investigación se sigue la idea de que al aprovechar los resultados más destacables del método autosupervisado propuesto por SimCLRv2 en conjunto de imágenes astronómicas, se obtendrá una red neuronal profunda capaz de identificar las características morfológicas generales presentes en dichas imágenes. Esto tiene un valor significativo en el ámbito astronómico, pues el aprendizaje de dichas representaciones generales depende únicamente de la cantidad de imágenes disponibles, que como ya se mencionó, son abundantes en esta área. Además, gracias a la disponibilidad de datos etiquetados, al realizar un ajuste fino sobre la red preentrenada de manera autosupervisada, se obtendrá un modelo clasificador de cinco tipos de galaxias. Se espera que el rendimiento de este último supere los resultados presentados por investigaciones relacionadas enfocadas únicamente en datos etiquetados, ya que se ha probado que el método autosupervisado permite el aprendizaje de las representaciones generales de gran calidad.

Al diferenciar los conjuntos de datos astronómicos y el conjunto de datos *ImageNet* (empleado en SimCLR), se deriva una segunda hipótesis: al extender el procesamiento de imágenes durante la fase autosupervisada a través de un conjunto de transformaciones que intensifique o resalte algunas de las características morfológicas de índole astronómica, las representaciones aprendidas serán de mayor calidad permitiendo al modelo semisupervisado una mejor detección de cada una de las clases a estudiar.

Puesto que las imágenes astronómicas obtenidas por los diversos estudios del cielo son en su mayoría de carácter desproporcionado en cuanto a su morfología (desbalance de tipos de galaxias), es altamente probable que independientemente del esquema de clasificación a considerar, los conjuntos de datos disponibles presenten un fuerte sesgo hacía las clases más representadas. De está problemática, a la cual se suma una pequeña cantidad de imágenes etiquetadas, se cree que un método que permita generar pseudoetiquetas a lo largo del entrenamiento autosupervisado, permitirá al modelo generar representaciones generales de mayor calidad con respecto al método original para las clases más desbalanceadas. Esta hipótesis se encuentra fundamentada en el funcionamiento del método autosupervisado propuesto por SimCLR, pues éste se basa en la separación espacial de las representaciones consideradas negativas y la aglomeración de aquellas consideradas como positivas.

Bajo la observación de que codificadores más recientes y con estructuras más complejas, el rendimiento de las redes neuronales suelen superar el rendimiento de clasificación con respecto a los modelos más antiguos y, por ende, menos complejos, se conjetura que al cambiar el codificador propuesto por SimCLR por uno más reciente y que ha probado superar el rendimiento de éste, mejorará la clasificación morfológica de galaxias con respecto al método SimCLR entrenado para el mismo fin.

Como es de esperar, la clasificación morfológica de galaxias se basa fuertemente en la detección de regiones relevantes sobre el objeto de interés así como su estructura. Por consiguiente algunas técnicas de visualización que permitan reconocer los patrones aprendidos por algunos modelos como lo son las activaciones fuertes por capas. También responder a las preguntas ¿cómo se descompone una imagen a través de las diferentes filtros de dichos modelos? (cuadriculas de activación) y ¿qué características aportan mayor valor a la predicción? (valores SHAP) son de gran valor, ya que permitirán generar algunos hitos visuales de la clasificación morfológica de galaxias que podrían ser útiles para extender esta o futuras investigaciones relacionadas.

1.6. Aportes

Las aportaciones realizadas durante esta investigación pueden resumirse en:

- La implementación de la arquitectura SimCLRv2, considerada estado del arte del aprendizaje semisupervisado, para la clasificación morfológica de galaxias.
- La obtención de un conjunto de transformaciones de preprocesamiento de imágenes que faciliten el aprendizaje de las representaciones del dominio particular.
- Brindar una propuesta para atacar el problema del desbalance de clases inherente a la clasificación morfológica de galaxias.
- Análisis del impacto en la tarea de clasificación de cinco clases de galaxias al considerar los dos puntos anteriores.
- Ofrecer un análisis cualitativo y visual para la comprensión de regiones de clasificación, distribución real e inferencia de clases y la demostración de la cantidad de clases óptima para el conjunto de datos empleado.

1.7. Organización del presente trabajo

El presente trabajo se divide en siete capítulos, en los cuales se procura dar una descripción breve y concisa del tópico a tratar.

En el capítulo 2 se describen los conceptos básicos y avances del aprendizaje profundo aplicados al procesamiento y clasificación de imágenes. Se realiza una distinción entre las técnicas de aprendizaje supervisado, autosupervisado y transferencia de conocimiento.

Gracias al contenido del capítulo 2 es posible discutir, de una manera más natural, acerca de la arquitectura del estado del arte del aprendizaje semisupervisado para la clasificación de imágenes, la cual será el punto de partida para los experimentos propuestos, esto se realiza en el capítulo 3.
Una vez comprendidos los conceptos básicos y la arquitectura del estado del arte, en el capítulo 4 se dará a conocer la intención y propuesta de esta investigación, así como una descripción del conjunto de datos utilizado durante todos los experimentos.

Durante el capítulo 5 se detallará, de forma exhaustiva, los experimentos realizados y los parámetros utilizados.

A lo largo del capítulo 6 se mostrarán los resultados obtenidos para cada experimento planteado en el capítulo 5 y se ofrecerá una discusión en torno a ellos.

Finalmente, en el capítulo 7 se abordarán las conclusiones con base en la discusión generada en el capítulo anterior y se propondrán, como trabajo a futuro, posibles modificaciones a los métodos y/o a la arquitectura con el fin de ampliar el contenido de esta investigación.

Capítulo 2

Fundamentos de redes neuronales

2.1. Neurona artificial y el perceptrón

Las redes neuronales constan de un conjunto de unidades de procesamiento básicas, que llamamos neuronas artificiales, las cuales se encuentran interconectadas entre sí. Estas neuronas artificiales imitan, de una forma muy limitada, el funcionamiento de los axones o neuritas presentes en el cerebro biológico. En la neurona biológica, los cuerpos receptores, llamados dendritas, reciben señales de varias neuronas vecinas, estas señales son procesadas por el cuerpo de la neurona receptora, para posteriormente ser enviadas al axón de la neurona. Si la señal captada por el axón supera un cierto umbral, éste emitirá una nueva señal hacia las dendritas de otras neuronas.

Las neuronas artificiales fueron presentadas por primera vez por McCullogh y Pitts en 1943 [13], quienes, para su creación, realizaron una simplificación del modelo biológico al plantear que todas las conexiones sinápticas se localizaban directamente en el soma de la neurona. Así como las dendritas, las conexiones de entrada en una neurona artificial transmiten señales de otras neuronas vecinas. Las señales recibidas son sumadas para posteriormente emitir una señal basada en un umbral predefinido. Por ejemplo, en el modelo de McCullogh y Pitts, la señal emitida podía tomar los valores 0 o 1. En general, la única distinción entre otros tipos de neuronas artificiales es la señal de salida producida por éstas. Una manera muy natural de ampliar o modificar el comportamiento de esta señal de salida, consiste en la aplicación de una función f sobre la suma de las señales de entrada. A f se le conoce como función de activación. La figura 2-1 muestra la representación de una neurona artificial con sus equivalencias biológicas marcadas con flechas.

McCullogh y Pitts propusieron una serie de neuronas con las cuales sería posible implementar todas las funciones Booleanas. Sin embargo, estas implementaciones no contaban con un método de aprendizaje por lo que era necesario ajustar manualmente las señales de entrada (pesos w y sesgos b) y realizar combinaciones de neuronas para lograr una implementación correcta [15]. No fue hasta 1957 que Rosenblatt y colaboradores propusieron el Perceptrón, cuyas unidades básicas eran muy similares a las neuronas artificiales presentadas por McCullogh y Pitts. Para Rosenblatt, la finalidad del Perceptrón era su aplicación en tareas de clasificación binarias, por lo que proporcionó al Perceptrón un método de aprendizaje mediante el cual los pesos y sesgos de cada neurona podían ser modificados automáticamente a través de una serie de reglas específicas [14].

No obstante, a finales de los 60's Minsky y Seymour, probaron que el algoritmo de aprendizaje de Rosenblatt presentaba una serie de limitaciones al implementar funciones booleanas no lineales, como XOR y XNOR. Debido a esto, gran parte de la investigación en redes neuronales se detuvo durante una década, a este periodo se le conoce como *el primer invierno de la IA* [15].

El primer invierno de la IA terminó, a finales de los 80's, gracias a la implementación del algoritmo de retro-

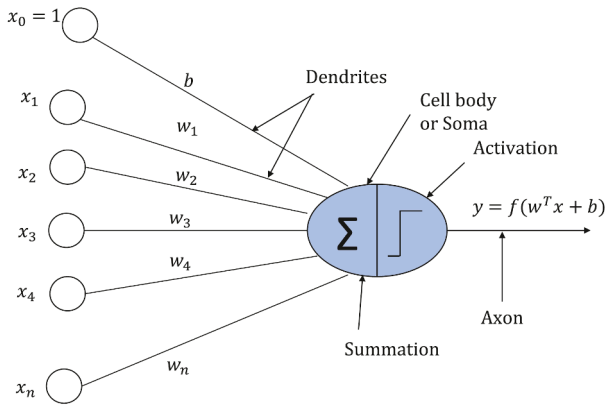


Figura 2-1: El modelo de la neurona artificial y sus equivalencias con la neurona biológica. Imagen tomada de [14].

propagación como un método de aprendizaje para las redes neuronales. En [16] Hinton, Rumelhart y Williams probaron que el algoritmo de retropropagación permitía a las redes neuronales resolver problemas que se había demostrado eran insolubles con el algoritmo de aprendizaje de Rosenblatt. Incluso en la actualidad este algoritmo es una de las piezas esenciales en el aprendizaje de las redes neuronales.

2.2. Perceptrón multicapa

En la sección anterior se explicó, intuitivamente, el funcionamiento de una neurona artificial. Más formalmente, la definimos como una función matemática que recibe $n+1$ valores reales (ver fig 2-1) y regresa un valor real, es decir una neurona es una función $\psi(\vec{x}) : \mathbb{R}^{n+1} \to \mathbb{R}$, definida como:

$$\psi(\vec{x}) = f\left(\sum_{i=1}^{N} w_i x_i + b\right) = f\left(\mathbb{W} \cdot \vec{x} + b\right)$$

donde f es la función de activación, $f(y) : \mathbb{R} \to \mathbb{R}$ y w_i, x_i son los i-ésimos pesos y entradas respectivamente. Para la igualdad derecha, $\mathbb{W}$ denota la matriz transpuesta de pesos. Para este caso $\mathbb{W}$ es un vector n dimensional.

Un perceptrón multicapa es por definición una red neuronal que consiste de un arreglo ordenado (en forma de columnas) de, al menos tres columnas, que llamamos capas, de neuronas artificiales interconectadas entre sí. La cantidad de neuronas en la primer capa, dependerá del problema a resolver, así como a las características de los datos disponibles para el mismo, a esta primer capa se le conoce como *capa de entrada*. En contraparte, la cantidad de neuronas de las siguientes capas, excepto la última, son de libre elección y se conocen como *capas ocultas*. La última capa, llamada *de salida*, contendrá una cantidad de neuronas determinada por el problema a resolver. Cuando se dice que la red es un arreglo ordenado, nos referimos a que las salidas de las neuronas de la capa de entrada serán las entradas de la primer capa oculta, cuyas salidas a su vez serán entradas de la segunda capa oculta, hasta llegar a la capa de salida (Fig. 2-2). Es por esta razón por la que se suele llamar a estas redes neuronales como *redes neuronales prealimentadas* o *feedforward neural networks*.

La figura 2-2 nos permite introducir la notación del perceptrón multicapa, que será útil para detallar el algoritmo de retropropagación. Entonces, de la fig. 2-2 se deduce que: w^{ℓ}_{jk} denota el peso de la k-ésima neurona de la capa $\ell - 1$ a la j-ésima neurona en la capa ℓ, a^{ℓ}_{j} es la función de activación aplicada a la suma Σ en la j-ésima neurona de la capa ℓ. De la misma forma, denotamos el sesgo de la j-ésima neurona en la capa ℓ como b^{ℓ}_{j}. Por otro lado, Σ es la abreviación de $\mathbb{W} \cdot \vec{x} + b$.

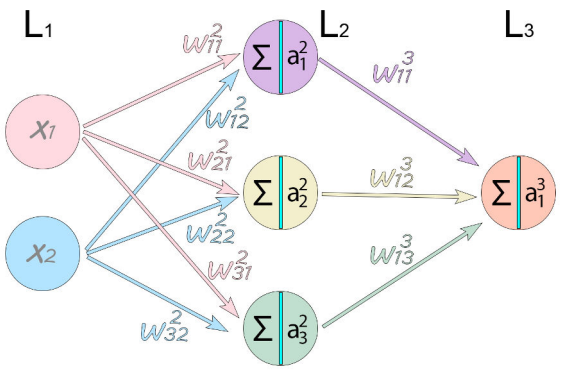


Figura 2-2: Perceptrón multicapa con dos neuronas en la capa de entrada L_1, tres neuronas en la capa oculta L_2 y una neurona en la capa de salida L_3. Cada neurona, a partir de la capa oculta, realiza la operación suma Σ y a la cual se aplica la función de activación a_j^ℓ. Mientras que w_{12}^2 denota el peso de la segunda neurona de la capa 1 a la primera neurona en la capa 2.

Con esta notación, es posible inferir el valor de la función de activación en la capa ℓ, a partir de la matriz de pesos $\mathbb{W}^\ell$ y el valor de la función de activación en la capa $\ell - 1$, mediante la ecuación:

$$a^\ell = f\left(\mathbb{W}^\ell \cdot \vec{a}^{\ell-1} + \vec{b}^\ell\right) = f\left(\Sigma^\ell\right) \tag{2-1}$$

2.3. Algoritmo de retropropagación

El algoritmo de retropropagación permite calcular el gradiente de la función de pérdida o costo el cual posibilita emplear el algoritmo del descenso por gradiente mediante el cual es posible actualizar, de forma automática, los parámetros contenidos en la matriz de pesos $\mathbb{W}^\ell$ y los vectores de sesgo $\vec{b}^\ell$. La actualización de dichos pesos es de tal manera que se busca minimizar función de costo $C(\mathbb{W}, \vec{b})$ que es una función dependiente del problema a resolver.
Por ejemplo, si se tratase con el problema de regresión, la función de perdida estaría dada por $C = \frac{1}{2N}\sum_x ||y(x) - a^L(x)||_2$, donde N es la cantidad total de muestras de entrenamiento, $y(x)$ es el vector que contiene los valores de la muestras x, $a^L(x)$ es el vector de activaciones de salida de la red neuronal cuando x es la entrada y $||\cdot||_2$ es la norma euclídea.

En el algoritmo de retropropagación, el cálculo de las derivadas parciales de C con respecto a los parámetros $\mathbb{W}$ y $\vec{b}$ se da de la capa de salida hacía atrás. Para entender dicho cálculo, considérese un perceptrón multicapa con L capas y sea ∂_j^ℓ el error de la j-ésima neurona en la capa ℓ, dado por:

$$\partial_j^\ell = \frac{\partial C}{\partial \Sigma_j^\ell} \tag{2-2}$$

Para calcular el error de la j-ésima neurona en la última capa L, veamos que la salida está dada por $a_j^L = f\left(\Sigma_j^L\right)$, de esta forma, el error de salida será:

$$\partial_j^L = \frac{\partial C}{\partial a_j^L}\frac{d}{d\Sigma_j^L} f\left(\Sigma_j^L\right) \tag{2-3}$$

Podemos reescribir la ecuación 2-3 en términos vectoriales de la siguiente manera:

$$\partial^L = \vec{\nabla}_a C \odot \frac{d}{d\Sigma^L} f\left(\Sigma^L\right) \tag{2-4}$$

Donde $\odot$ denota el producto de Hadamard.
Puede demostrarse, al manipular la notación aquí propuesta, que el error para cualquier capa ℓ está relacionado con el error de la capa subsecuente, es decir:

$$\partial^\ell = \left(\left(\mathbb{W}^{\ell+1}\right)^\intercal \partial^{\ell+1}\right) \odot \frac{d}{d\Sigma^\ell} f\left(\Sigma^\ell\right) \tag{2-5}$$

Por otro lado, la razón de cambio de C con respecto al peso de la neurona j en la capa ℓ, se encuentra relacionada con el error de la misma. Para probar esto, partimos de la derivada de C con respecto a w^ℓ_{jk} y se descompone usando la regla de la cadena:

$$\frac{\partial C}{\partial w^\ell_{jk}} = \frac{\partial C}{\partial \Sigma^\ell_j}\frac{\partial \Sigma^\ell_j}{\partial w^\ell_{jk}} = \partial^\ell_j \frac{\partial \Sigma^\ell_j}{\partial w^\ell_{jk}} \tag{2-6}$$

Ahora bien, por definición se sabe que $\Sigma^\ell_j = \sum_{j=1}^N w^\ell_{jk} a^{\ell-1}_k + b^\ell_j$, luego derivando con respecto a w^ℓ_{jk} se tiene que:

$$\frac{\partial \Sigma^\ell_j}{\partial w^\ell_{jk}} = a^{\ell-1}_k \tag{2-7}$$

De esta forma se sigue que la derivada parcial de C con respecto a los pesos puede escribirse como:

$$\frac{\partial C}{\partial w^\ell_{jk}} = a^{\ell-1}_k \partial^\ell_j \tag{2-8}$$

La razón de cambio de C con respecto a los sesgos de cada neurona también están relacionados con el error de las mismas. De una manera análoga al procedimiento anterior, se demuestra que dicha relación se expresa como:

$$\frac{\partial C}{\partial b^\ell_j} = \partial^\ell_j \tag{2-9}$$

Las ecuaciones fundamentales del algoritmo de retropropagación son: 2-4, 2-5, 2-8 y 2-9 que, como ya se mencionó, permiten obtener el gradiente de la función de costo. En el algoritmo 1 se muestra el pseudocódigo del mismo.

Una vez obtenido el gradiente de la función de costo, la actualización de los pesos y sesgos debe seguir un método iterativo de optimización sobre C. El método más sencillo es el descenso por gradiente, cuya actualización de parámetros es de la forma:

$$\mathbb{W}^\ell_{t+1} := \mathbb{W}^\ell_t - \eta \frac{\partial C}{\partial \mathbb{W}^\ell} \tag{2-10}$$

$$\vec{b}^\ell_{t+1} := \vec{b}^\ell_t - \eta \frac{\partial C}{\partial \vec{b}^\ell} \tag{2-11}$$

donde, η es un hiperparámetro llamado tasa de aprendizaje y $t \in \mathbb{N}$ representa el paso de entrenamiento actual. Nótese que el algoritmo de retropropagación proporciona el gradiente de C para un paso de entrenamiento, por lo que éste también suele aplicarse de forma iterativa, donde a cada iteración se le llama paso de entrenamiento.

Algoritmo 1 Algoritmo de retropropagación en redes neuronales

Entrada: Red Neuronal con pesos y sesgos $\mathbb{W}$ y $\vec{b}$, conjunto de entrenamiento $\vec{x}$, función C
Salida: El gradiente de la función de costo C

1: Establece la función de activación a^1 para la capa de entrada en función de las entradas $\vec{x}$.
2: Propagación: Para cada $l = 2, 3, ..., L$ calcula $\Sigma^\ell = \mathbb{W}^\ell \vec{a}^{\ell-1} + \vec{b}^\ell$ y $\vec{a}^\ell = f\left(\Sigma^\ell\right)$
3: Calcula el error de salida $\partial^L = \vec{\nabla}_a C \odot \frac{d}{d\Sigma^L} f\left(\Sigma^L\right)$ (ecuación 2-4)
4: retropropaga el error: Para cada $l = L-1, L-2, ..., 2$ calcula $\partial^\ell = \left(\left(\mathbb{W}^{\ell+1}\right)^\intercal \partial^{\ell+1}\right) \odot \frac{d}{d\Sigma^L} f\left(\Sigma^L\right)$ (ecuación 2-5)
5: Regresa: El gradiente de la función de costo dado por $\frac{\partial C}{\partial w^\ell_{jk}} = a^{\ell-1}_k \partial^\ell_j$ y $\frac{\partial C}{\partial b^\ell_j} = \partial^\ell_j$ (ecuaciones 2-8 y 2-9)

A pesar de que el descenso por gradiente es una técnica extremadamente sencilla, no siempre es efectiva, ya que en algunas ocasiones ésta presenta problemas de divergencia entre otros. Por estas razones se han estudiado una diversidad de técnicas de optimización, como por ejemplo: *AdaGrad* [17], *Momento*, *RMSProp*, *Adam* [18], *LARS* [19], entre otros.
Dado que los experimentos propuestos en este trabajo emplearon la técnica LARS, es conveniente presentar su método de actualización de pesos (ver el algoritmo 2).

Algoritmo 2 Algoritmo LARS

Entrada: tasa de aprendizaje η_t, conjunto de entrenamiento $\vec{x}$, función de escala ϕ, parámetro $0 < \beta < 1$, función de Costo C
Salida: Parámetros $\mathbb{W}^\ell$ y $\vec{b}^\ell$ actualizados

1: Define $m_0 = 0$
2: **for** t=1 **hasta** T **do**
3: Calcula $g_t = \frac{1}{N} \sum^N \vec{\nabla} C$
4: Calcula $m_t = \beta m_{t-1} + (1-\beta) g_t$
5: Calcula $\mathbb{W}^\ell_{t+1} = \mathbb{W}^\ell_t - \eta_t \frac{\phi\left(||\mathbb{W}^\ell_t||\right)}{||m^\ell_t||} m^\ell_t \; \forall \ell \in L$
6: **end for**

2.4. Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales o *CNN* (por las siglas en inglés de *Convolutional Neural Networks*) surgieron de la necesidad de resolver problemas en el área de la visión computacional, ya que se había encontrado que el perceptrón multicapa no lograba modelar correctamente una gran cantidad de dichos problemas. Dado que el perceptrón provenía de la idea de modelar el cerebro biológico, las CNN fueron construidas bajo el mismo concepto; se buscaba simular o replicar la visión humana [15].

Con el fin de explicar el funcionamiento del procesamiento de la información visual en el cerebro humano, Hubel y Wiesel desarrollaron un método para examinar los campos receptivos de algunas células pertenecientes a la corteza visual en gatos. Gracias a su estudio [20], los autores reconocieron dos tipos de células, que llamaron simples y complejas, las cuales juegan un papel fundamental para el reconocimiento de patrones en los objetos. Por un lado, las células simples responden a bordes o contornos y barras de orientaciones particulares. Mientras que las células complejas también responden a bordes o contornos y barras de orientaciones particulares, con la diferencia de que éstas poseen la propiedad de invariabilidad de posición en el campo receptivo, es decir, la respuesta de la célula no depende de la posición del objeto. Hubel y Wisel propusieron que la propiedad invariante de posición se lograba gracias a la información procesada de un conjunto de células simples que responden a las mismas orientaciones pero con diferentes posiciones del campo receptivo.

Uno de los primeros modelos computacionales, inspirados en replicar la visión humana, fue el *Neocognitron* de Fukushima [15], quien basado en los trabajos de Hubel y Wiesel dotó al Neocognitron de dos tipos de capas; las capas S como extractoras de características y las capas C como conexiones estructuradas para organizar las características extraídas. Las capas S consisten de un determinado número de "células simples" para modelar (junto con las capas C) el comportamiento de las células complejas. Idealmente estas "células simples" pueden ser entrenadas para responder a una característica particular en su campo receptivo. En general, en el Neocognitron, la extracción de características locales como bordes en determinadas orientaciones, se da en las capas iniciales o capas inferiores, mientras que la extracción de las características globales se produce en las capas superiores [15].

El modelo del Neocognitron fue la fuente de inspiración para la construcción de las redes neuronales convolucionales, que, en la actualidad, están constituidas por tres grandes bloques:

1. Capas convolucionales: permiten la extracción de características a través de una operación matemática llamada convolución, la cual es muy similar a la extracción realizada por las "celulas simples" de Fukushima.

2. Capas de submuestreo: suelen colocarse inmediatamente después de una capa convolucional, ya que a través de éstas es posible reducir el número de parámetros entrenables de la red.

3. Perceptrón multicapa o capas completamente conectadas: sus entradas son las características extraídas por las capas anteriores para su clasificación.

En las siguientes subsecciones se dará una descripción más detallada de cada una de estas tres capas.

2.4.1. Capa convolucional

La operación de convolución es una operación matemática entre dos funciones f y g que produce una tercera función s, que puede ser interpretada como la interpolación entre dos señales (f y g) que cuantifica el impacto de una señal cuando ésta se integra sobre la otra señal. Esta operación se define formalmente para el caso continuo y discreto como:

$$s(t) = (f \star g)(t) = \int_{-\infty}^{\infty} f(k)g(t-k)dk = \sum_{k=-\infty}^{\infty} f(k)g(t-k) \tag{2-12}$$

En las CNN, $f(k)$ son los datos de entrada, que para la primera capa, es la representación matemática de una imagen, g representa lo que se conoce como filtro o *kernel* y a $s(t)$ se le llama mapa de características.

Una imagen puede ser representada matemáticamente como un conjunto acoplado de matrices o tensores, en las cuales cada elemento contiene un valor $a \in \mathbb{R}$ o $\mathbb{N}$ que suele llamarse píxel. Cada matriz, acoplada en este conjunto se conoce como canal de color (C_i). Los valores que puede adquirir cada píxel en cada canal de color dependen del modelo de color con el cual se trabaja. Por ejemplo, en el modelo de color rojo, verde y azul (RGB), cada píxel puede tomar valores en el intervalo $[0, 255]$, aunque para evitar problemas de sobreajuste, debido a la variabilidad de este intervalo, las imágenes se normalizan al intervalo $[0, 1]$. Por otro lado, la cantidad de filas y columnas del conjunto acoplado define el ancho (w) y alto (h) de la imagen respectivamente.

Un filtro, en las CNN, son típicamente matrices multidimensionales de parámetros entrenables $\mathbb{W}$ de dimensiones $m \times n \times D$, donde n, m corresponden al alto y ancho del filtro, tal que n, m son menores a las dimensiones h, w de la entrada y D es menor o igual a los canales de entrada C_i. El proceso de entrenamiento de estos filtros es relativamente simple; en primer lugar, debe elegirse la cantidad y dimensiones a aplicar en cada capa convolucional. Al inicio del entrenamiento, los valores de cada filtro suelen inicializarse de manera aleatoria. Durante la propagación hacia adelante del algoritmo de retropropagación, cada filtro se convoluciona a cada posible píxel en la imagen de entrada para generar un mapa de características. Este mapa de características actuará, ahora, como una imagen de entrada para las capas convolucionales subsecuentes que conducen a

la extracción de características de alto nivel.

Para clarificar la operación de convolución, supongamos por simplicidad una imagen G en escala de grises, es decir $C_i = 1$, de alto y ancho h_1, w_1 respectivamente y un filtro K de dimensiones $m_1 \times n_1 \times 1$. Puesto que tanto la imagen G como el filtro K son finitas y discretas, el valor de la operación de convolución de K sobre G en las posiciones i, j se expresa como:

$$s(i,j) = \sum_{m=0}^{m_1} \sum_{n=0}^{n_1} G(m,n) K(i-m, j-n) \tag{2-13}$$

La figura 2-3 muestra el resultado de la convolución de un filtro de dimensión 3×3 sobre una imagen de dimensiones $5 \times 5 \times 1$.

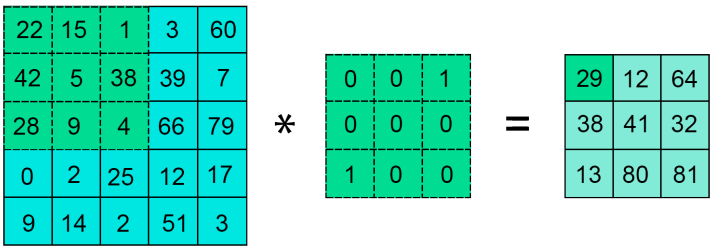


Figura 2-3: Convolución aplicada a una matriz de 5×5 con un filtro de 3×3. El resultado es una matriz de 3×3 pues el recorrido del filtro se encuentra ajustado a un píxel por paso.

Un aspecto importante a considerar es el tamaño $k_1 \times k_2$ de los filtros, la cantidad de los mismos, así como el paso o recorrido (*stride*) st sobre la imagen, ya que las dimensiones del mapa de características dependen de dichos valores. En la imagen 2-3, el tamaño del *kernel* es $k_1 = k_2 = 3$, y el paso del recorrido $st = 1$ en ambas direcciones. Si se cambiase el valor del recorrido en ambas direcciones por $st = 2$, el mapa de características tendría dimensiones de 2×2. Por otro lado, si se cambiase el valor del recorrido por $st = 3$, la operación de convolución no está bien definida. Para evitar esto, se suele agregar a la imagen de entrada, una cantidad determinada de filas y columnas con valor cero en sus entradas. A este procedimiento se le conoce como *zero padding* y se denota como p_j, donde j es la cantidad de filas y columnas añadidas. La figura 2-4 muestra el mismo ejemplo que en la figura 2-3 con $k_1 = k_2 = 3$, p_1 y $s_t = 2$; el mapa de características es también de dimensiones 3×3.

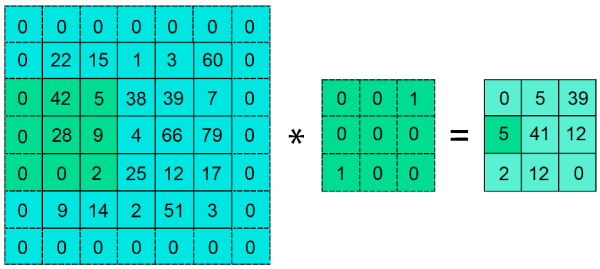


Figura 2-4: Convolución aplicada a una matriz de 5×5, con *zero padding* p_1, mediante un filtro de 3×3 y *stride* st de dos. El resultado es una matriz de 3×3.

Puesto que los valores de los filtros convolucionales en una CNN no son más que parámetros entrenables, el diseño de éstos es un paso extremadamente importante para garantizar un buen desempeño de la red.

2.4.2. Capas de submuestreo

Las capas de agrupamiento (*pooling layers*) o capas de submuestreo (*subsampling layers*) son comúnmente aplicadas inmediatamente después de cada capa convolucional. Su principal función consiste en reducir las dimensiones espaciales de los mapas de características, permitiendo una reducción en los parámetros entrenables de la red. Adicionalmente, se ha demostrado que estas capas reducen la probabilidad de sobreajuste en la red [21].

La reducción de las dimensiones de los mapas de características, se realiza a través de algunas funciones que permitan sintetizar la información de entrada en determinadas subregiones de la misma, siendo las operaciones más comunes el promedio y la función máximo. Este proceso sigue la misma idea que la operación de convolución, salvo que, se reemplaza la combinación lineal por dicha función sintetizadora. La figura 2-5 muestra el resultado de aplicar las operaciones de máximo y promedio a una imagen de entrada.

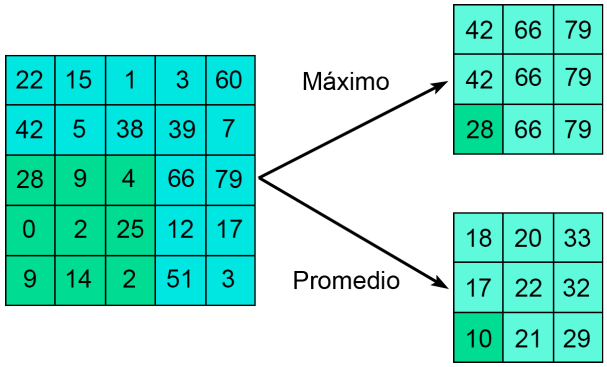


Figura 2-5: Dos capas de submuestreo, a través de la convolución y las funciones sintetizadoras máximo y promedio. Para ambos casos el filtro se aplica con *stride* st de uno.

Al igual que en las capas de convolución, el tamaño de las capas de submuestreo depende del tamaño de la entrada $m \times n$, el tamaño del "filtro" $k_1 \times k_2$ que recorrerá la imagen y el valor del paso de recorrido st.

2.4.3. Capas completamente conectadas

Comúnmente, el último bloque de las CNN, suele consistir en un perceptrón multicapa, al cual se le refiere como capas completamente conectadas o capas densas, ya que las neuronas en estas capas se encuentran conectadas a todas las activaciones de la capa anterior. La salida de las capas densas es usualmente, una puntuación de clases, donde el número de neuronas iguala al número de clases.

De esta forma, la combinación de las capas convolucionales, de submuestreo y completamente conectadas posibilitan la transformación de una imagen de entrada en una representación numérica que permite su clasificación.

2.5. Arquitecturas ResNet-Ω

El *Large Scale Visual Recognition Challenge* o ILSVRC por sus siglas en inglés, es uno de los criterios de referencia para la evaluación del rendimiento de las arquitecturas CNN en las tareas de reconocimiento de objetos a gran escala [22]. Esta competencia, celebrada cada año desde el 2010, ha promovido la creación y perfeccionamiento, acelerados, de nuevas arquitecturas, que año tras año han demostrado una reducción significativa en el error de clasificación. Una de las primeras arquitecturas en mejorar el error de clasificación con respecto al humano, fue la arquitectura ResNet-152, la cual en el año 2015 obtuvo un error del 3.57 %, una mejora relativa

de 1.43 % con respecto al rendimiento humano [23].

Una de las principales ideas en la tarea de clasificación de imágenes a gran escala, consiste en el tamaño de la arquitectura, se sabe que entre más profunda es la red, mejor será su rendimiento. Sin embargo, este procedimiento no es, ni lo fue, sencillo de implementar, pues al incrementar la cantidad de capas, se observa el fenómeno del desvanecimiento del gradiente, en el cual, las primeras capas de la red no se actualizan de ninguna manera (el gradiente es cero) o su actualización es extremadamente lenta (el gradiente es muy pequeño), comprometiendo de esta manera el aprendizaje de la red.

Las redes neuronales residuales o redes ResNet fueron presentadas por He y colaboradores, quiénes encontraron que además del problema del desvanecimiento del gradiente, las redes cada vez más profundas presentaban el problema del *deterioro de la exactitud*. En éste la exactitud se satura en un determinado punto para después degradarse rápidamente. Los autores argumentaron y probaron experimentalmente, que este problema no se debía al sobreajuste, si no más bien a una pobre optimización de las arquitecturas profundas, pues teóricamente el error de clasificación de una red más profunda, es por construcción, mucho menor que su contraparte menos profunda [24].

Para atacar los problemas del deterioro de la exactitud y el desvanecimiento del gradiente en las redes profundas, los autores propusieron el aprendizaje profundo residual; un marco de trabajo en el cual se introduce el mapeo residual de capas convolucionales apiladas $\mathcal{H}(\vec{x})$, definido como:

$$\mathcal{H}(\vec{x}) = \mathcal{F}(\vec{x}) + \vec{x} \tag{2-14}$$

donde, $\vec{x}$ denota la entrada de las primeras capas apiladas y $\mathcal{F}(\vec{x})$ es la salida de las capas apiladas, llamada *función residual*. Por definición, el mapeo residual no añade parámetros ni complejidad computacional extra.

Suponiendo que un conjunto de capas no lineales aproximan asintóticamente funciones complejas, entonces el conjunto de capas apiladas pueden aproximar la función residual $\mathcal{F} = \mathcal{H} - \vec{x}$. La figura 2-6 muestra el mapeo $\mathcal{H}$ o conexión residual.

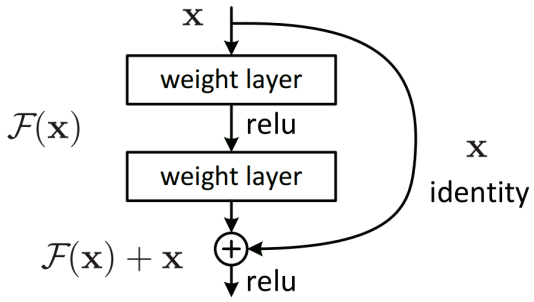


Figura 2-6: Conexión o Mapeo residual en dos capas apiladas. relu denota la función de activacion Rectified Linear Unit (ReLU). Imagen tomada de [24].

Un conjunto de capas apiladas con su respectivo mapeo residual define un bloque residual, que de acuerdo a la cantidad de capas convolucionales, el tamaño del filtro, el tamaño de paso st y la cantidad de veces que se aplica dicho bloque, definen la cantidad de capas Ω en la arquitectura, la cual se denomina Resnet-Ω.

Para cualquier Resnet-Ω, la capa convolucional de entrada tiene dimensiones de $7 \times 7 \times 64$ en el tamaño del *kernel*, con $st = 2$, seguida por una capa de submuestreo máximo con tamaño 3×3 y $st = 2$. A esta primer capa convolucional no se emplea el mapeo convolucional, por lo que no se considera bloque residual.

La figura 2-7 muestra los bloques residuales fundamentales de las arquitecturas Resnet-50,-101,-152, según la cantidad de veces que se aplica a cada bloque (*cfg* en la imagen), mientras que la imagen 2-8 muestra los bloques residuales fundamentales de las arquitecturas Resnet-18,-34.

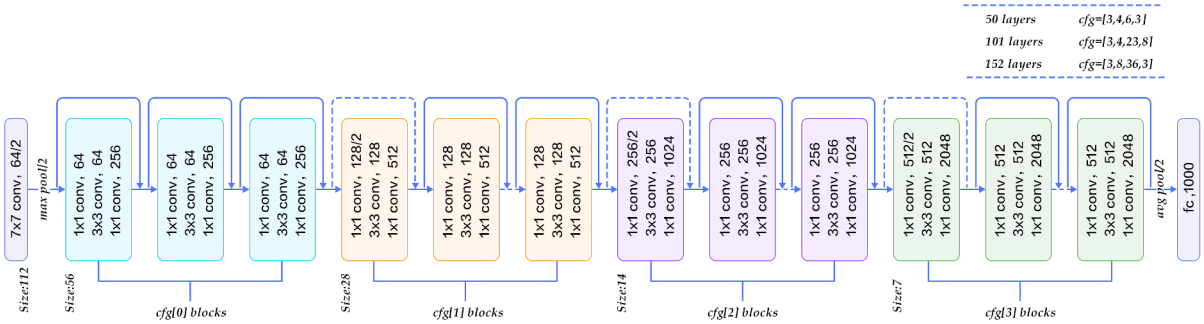


Figura 2-7: Arquitecturas Resnet-50,-101,-152. La cantidad de veces que se repite cada bloque residual determina cada arquitectura. Por ejemplo, Resnet-50 consta de tres bloques $cfg[0]$ (azul), cuatro bloques $cfg[1]$ (amarillo), seis bloques $cfg[2]$ (morado) y tres bloques $cfg[3]$ (verde). Imagen tomada de `raw.githubusercontent.com/PaddlePaddle/book/develop/03.image_classification/image/resnet.png`.

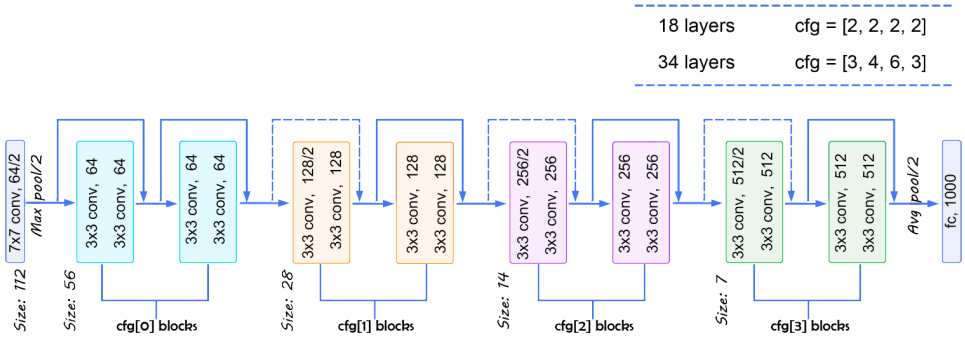


Figura 2-8: Arquitecturas Resnet-18,-34. La cantidad de veces que se repite cada bloque residual determina cada arquitectura. Las líneas punteadas denotan un aumento en las dimensiones de salida, por lo que la conexión residual debe aplicarse con *zero padding* o aplicar una proyección lineal de tal forma que la operación suma esté bien definida.

2.6. Aprendizaje autosupervisado

Hasta el día de hoy muchas de las aplicaciones exitosas del aprendizaje de máquinas en diversas áreas como la visión computacional, reconocimiento de voz, traducción y procesamiento del lenguaje natural han involucrado la técnica del aprendizaje supervisado, en el cual se emplean conjuntos de datos etiquetados cuyas etiquetas facilitan la minimización de las funciones de costo al comparar directamente las predicciones realizadas por la red con el valor real, en consecuencia, el rendimiento de la red se ve favorecido. Sin embargo, la generalización del conocimiento adquirido por estos sistemas se encuentra limitado a las etiquetas con las cuales fueron entrenados. Además, la creación de conjuntos de datos es una tarea sumamente complicada, que demanda una gran cantidad de recursos humanos y económicos.

En las últimas dos décadas, una gran cantidad de métodos del aprendizaje autosupervisado han sido estudiados y propuestos como una solución viable para eludir el costoso etiquetado de datos, pues en este método se

aspira al aprendizaje de las características de conjuntos a gran escala sin el uso de etiquetas producidas por humanos. Además de evitar el uso de conjuntos de datos etiquetados el aprendizaje autosupervisado es un método más robusto que su contraparte supervisada, siempre y cuando el primero esté correctamente modelado. Puesto a que en el método autosupervisado el problema a resolver no se encuentra limitado al uso de las etiquetas, el aprendizaje de las características del conjunto de datos es más general y por ende, el sistema será capaz de hallar patrones ocultos y cambiantes en el tiempo [25]. No obstante, el aprendizaje de representaciones útiles a través de la técnica autosupervisada es aún un problema abierto.

En la actualidad, los métodos autosupervisados pueden dividirse en tres tipos o enfoques:

1. Los *enfoques generativos* aprenden a generar o modelar píxeles del espacio de entrada. Sin embargo, este método es extremadamente costoso computacionalmente y no son necesariamente útiles para el aprendizaje de las representaciones. Algunas de las arquitecturas más conocidas que emplean este enfoque son las redes generativas adversarias (GAN's por las siglas en inglés de *Generative Adversarial Networks*).

2. Los *enfoques discriminativos* permiten a las redes la extracción y aprendizaje de las representaciones por medio de funciones objetivo similares a las usadas en el aprendizaje supervisado. A través de la solución de una o varias *tareas pretexto* la red aprende algunas representaciones, implícitas en dichas tareas, y además puede generar pseudoetiquetas en función de los atributos del conjunto de datos [26]. Algunas de las tareas pretexto son: colorear imágenes en escala de grises, "pintado" de imágenes, resolver rompecabezas de jigsaw, inferencia de posiciones relativas, entre otros. El diseño de muchas de estas tareas están basadas en heurísticas, lo cual limita la generalidad de las representaciones aprendidas [12].

3. Uno de los enfoques que ha tomado gran fuerza durante los últimos años es el *enfoque contrastivo*, que ha demostrado ser un enfoque robusto para la extracción de representaciones generales. El objetivo de este enfoque consiste en generar representaciones de los ejemplos o instancias de tal manera que las representaciones de ejemplos similares sean cercanas entre sí, mientras que aquellas instancias diferentes entre sí se encuentren separadas. En la siguiente subsección se detallará este enfoque aplicado a la visión computacional, pues es la base fundamental de los experimentos propuestos en este trabajo.

Uno de los métodos más comunes y ampliamente usados para la evaluación de la calidad de las representaciones aprendidas por los algoritmos autosupervisados es la evaluación lineal o *linear evaluation* [12, 27–31], en donde se entrena un clasificador lineal con las salidas de la red con pesos congelados, siendo la exactitud en un conjunto de prueba un indicador de la calidad de dichas representaciones.

2.6.1. Aprendizaje contrastivo autosupervisado

Para realizar una diferenciación entre instancias, el enfoque contrastivo explota las similitudes implícitas en los ejemplos a través de la generación de muestras positivas, negativas y de referencia de un conjunto de datos no etiquetado $D = x_1, x_2, ..., x_N$ de tamaño N, donde cada ejemplo x_j pertenece a una única clase, es decir, D consta de N clases. La creación de ejemplos de la misma clase, requiere de la generación de un par (v^a, v^+) a partir de una misma muestra x_j, donde v^a, v^+ son muestras de referencia y positiva respectivamente. Para crear una muestra de una clase diferente, se genera v^- a partir de x_i, tal que $x_i \neq x_j$. A esta muestra v^- se le llama negativa.

El marco de trabajo para caracterizar los enfoques contrastivos existentes consta de cinco partes [32], que a continuación se describen.

Aumento de datos

La principal función del aumento de datos consiste en la generación de las muestras positivas, negativas y de referencia. Si se define $TA = (ta_1, ..., ta_k)$ como un conjunto de procesos estocásticos de aumento de datos

aplicado en forma secuencial, entonces, es posible generar una muestra nueva v^i al aplicar TA sobre una muestra x_i. Nótese que v^i preserva, esencialmente, parte de la información contenida en x_i y TA permite generar múltiples muestras de la misma clase definida por x_i.

La generación del par (v^a, v^+) puede realizarse de una gran cantidad de maneras. Una de ellas consiste en aplicar TA a la muestra x_i dos veces, es decir, $v^a = TA(x_i)$, $v^+ = TA(x_i)$. Debido a la naturaleza estocástica de TA, tanto v^a como v^+ serán dos conjuntos de características distintos, pero similares entre sí. La muestra negativa v^-, se obtiene al aplicar TA a cualquier muestra x_j tal que $x_j \neq x_i$.

Codificador

El codificador es una función aproximadora f_θ, parametrizada por θ, la cual mapea las muestras de entrada v a un conjunto de vectores $\vec{r}$, denominados representaciones de v. Si v es una imagen de C canales, alto h y ancho w, entonces el mapeo del codificador es de la forma $f_\theta : \mathbb{R}^{C\times h\times w} \to \mathbb{R}^{k\times C}$, esto es, el codificador regresa k vectores de dimensión C que representan la entrada v. Cuando el codificador es una CNN, $\vec{r}$ es un conjunto de vectores de un mapa de características $m \in \mathbb{R}^{c\times h\times w}$

Extracción de las representaciones

La extracción de las representaciones es fundamental en el aprendizaje contrastivo, ya que éstas son comparadas entre sí a través de una medida de similitud, que permite definir y minimizar una función de costo asociada.

Usando la idea de muestras positivas, negativas y de referencia al igual que el codificador, se define la extracción de las representaciones positivas, negativas y de referencia como $\vec{r}^+ = f_\theta(v^+)$, $\vec{r}^- = f_\theta(v^-)$ y $\vec{r}^a = f_\theta(v^a)$ respectivamente. La extracción de las representaciones también puede darse de diferentes formas, una de ellas consiste en generar un vector de dimension d como la salida del codificador, de esta forma, cada representación será unidimensional $r \in \mathbb{R}$. Otra forma de efectuar la extracción de las representaciones consiste en comparar cada subconjunto $\vec{r}^a$ de la matriz de representaciones $\vec{r}$ con otro subconjunto $\vec{r}^-$ para generar múltiples puntajes negativos.

Medida de similitud

La medida de similitud permite comparar las representaciones a través de una función $\Phi(\vec{r}_i, \vec{r}_j)$ que mide la similitud entre dos representaciones $\vec{r}_i$ e $\vec{r}_j$. Algunas de estas funciones son el producto punto, la similitud coseno y algunas transformaciones bilineales.

Función de pérdida

La definición de la función de costo debe ser de tal forma que se considere la combinación de los puntajes positivos, $s^+ = \Phi(\vec{r}^a, \vec{r}^+)$, y los puntajes negativos, $s^- = \Phi(\vec{r}^a, \vec{r}^-)$. Y debe ser de tal forma en la que la minimización de ésta sea equivalente a maximizar el puntaje positivo y minimizar el puntaje negativo.

Existen una gran variedad de funciones de pérdida para este marco de trabajo, algunas de ellas son: *estimación contrastiva negativa* (NCE) usada en [33], *pérdida de tripletes* empleada en [34], *InfoNCE* introducida por [35] y *NT-Xent* usada en [12, 27].

2.7. Transferencia de conocimiento

Una de las grandes ventajas que ofrecen los algoritmos del aprendizaje profundo con respecto a los algoritmos tradicionales del aprendizaje de máquina es el rendimiento en una gran variedad de tareas en diversos campos de interés, así como la habilidad de éstos de aprender y extraer características complejas en una gran variedad de conjuntos de datos. Sin embargo, estas ventajas suelen ser marginales siempre y cuando el volumen

de los datos empleados para el entrenamiento sea significativamente grande. Como ya se discutió en secciones anteriores, el etiquetado de datos de gran volumen es sumamente complicado de realizar, por lo que en la actualidad existen una cantidad muy limitada de éstos, de esta forma, se encuentra una restricción de aplicar estos métodos a través de diferentes conjuntos de datos.

El método de la transferencia de conocimiento permite sortear esta limitante al emplear una red neuronal previamente entrenada en un conjunto de datos de gran volumen sobre una tarea similar a la de interés. Este método puede ser empleado en dos principales estrategias basadas en el dominio, la tarea a resolver y la disponibilidad de los datos [36, 37]:

- **Como extractor de características**: Esta estrategia permite reutilizar los pesos aprendidos durante la tarea original, bajo la restricción de que éstos no se modifican bajo ninguna circunstancia. Dado que la nueva tarea puede diferir con respecto a la cantidad de clases y/o la función objetivo, entre otras, suelen retirarse las capas de clasificación o regresión de la red original y adaptar nuevas según los requerimientos de la nueva tarea. Durante el entrenamiento de esta nueva tarea, se usa el nuevo conjunto de datos y se congelan los parámetros de la red recortada, es decir, durante el proceso de entrenamiento éstos no son actualizados por el algoritmo de retropropagación y únicamente los parámetros de las nuevas capas son actualizados en función de las características extraídas por la red congelada.
- **Como inicialización**: Esta estrategia también permite reutilizar los pesos aprendidos durante la tarea original. A diferencia de la estrategia previamente mencionada, en ésta los pesos de toda la red o un conjunto de ellos, pueden ser actualizados a través del algoritmo de retropropagación y la nueva función objetivo (si es el caso) al emplear el nuevo conjunto de datos. A esta técnica también se le conoce como ajuste fino. Si se opta por actualizar todos los parámetros, debe tomarse en cuenta el valor de la tasa de aprendizaje, pues, sí este valor es similar al valor con el cuál la red fue entrenada originalmente, la red podría sufrir una modificación de sus pesos alta, en consecuencia, el rendimiento podría verse afectado.

Se ha demostrado que este método, aplicado a la visión computacional, por cualquiera de las dos estrategias, es robusto y permite obtener rendimientos linealmente correlacionados con el rendimiento de los modelos, ampliamente usados, en ImageNet [38].

Capítulo 3

Estado del arte

3.1. Aprendizaje auto y semisupervisado

De acuerdo con Papers With Code, la arquitectura SimCLRv2 presentada en [27], es hasta el año 2022 el estado del arte para las tarea de clasificación de imágenes a través de la técnica semisupervisada en el conjunto de datos a gran escala ImageNet, con valores de exactitud top-1 de $80.9\,\%$ y $79.8\,\%$ respectivamente. Por su parte, la arquitectura SimCLR [12] obtiene un valor de exactitud top-1 de $76.5\,\%$ para la tarea autosupervisada.

Como su nombre lo indica, SimCLRv2 es una versión mejorada de SimCLR. Puesto que las modificaciones realizadas a SimCLR son menores, puede considerarse que ésta es un caso particular de SimCLRv2.

Para este trabajo son de especial interés ambas arquitecturas, pues fueron la base fundamental para todos los experimentos desarrollados. Es por esta razón que durante las siguientes subsecciones se detallarán todos los aspectos relevantes de ambas arquitecturas.

3.1.1. Nociones generales de SimCLR

A Simple Framework for Contrastive Learning of Visual Representations (SimCLR) es una red neuronal profunda, que debido a sus características puede ser clasificada dentro del marco de trabajo que caracteriza a los enfoques contrastivos [32]:

1. **Aumento de datos**: A cada imagen x_k dentro de un lote del conjunto de datos de entrada, se le aplican dos veces un conjunto estocástico TA de aumento de datos, con el fin de obtener un par positivo y de referencia, de tal forma que $v^{a_k} = TA(x_k)$ y $v^{+_k} = TA(x_k)$. Dado que en cada lote existen, al menos dos imágenes, la generación de pares negativos se da de una forma muy natural. Si $x_j \neq x_k$, entonces tanto v^{a_j}, como v^{+_j} serán pares negativos de v^{a_k} y v^{+_k}. El conjunto de aumento de datos consiste de las transformaciones de recorte y redimensionado aleatorios, fluctuación de color, escala de grises aleatoria y desenfoque gaussiano.

2. **Codificador**: La función aproximadora f_θ empleada en SimCLR es una red neuronal convolucional ResNet-Ω, con $\Omega \in \{50, 101, 152\}$ de anchura y profundidad variables, con o sin *kernels* selectivos.

3. **Extracción de las representaciones**: Las representaciones $\vec{r}$, obtenidas por el codificador, son 1-dimensionales con valor de 2048. Las representaciones positivas $\vec{r}^{\,+}$ y de referencia $\vec{r}^{\,a}$ provienen del par (v^+, v^a). Dado que el aumento de datos se da por lotes, se define el conjunto de representaciones R^- negativo con respecto a (v^+, v^a). Note que el tamaño de R^- es dos veces el tamaño del lote.

4. **Medida de similitud**: En SimCLR la medida de similitud se realiza sobre un espacio vectorial normado de dimensiones menores o iguales a $\vec{r}$, para ello, se definen $\vec{z}_i$ e $\vec{z}_j$ dados por $\vec{z} = f_\psi(\vec{r})$, tal que $f_\psi : \mathbb{R}^c \to \mathbb{R}^m$. Para posteriormente, aplicar la similitud coseno entre $\vec{z}_i$ e $\vec{z}_j$.

5. **Función de pérdida**: La función de costo se minimiza sobre las proyecciones $\vec{z}$ de las representaciones positivas, negativas y de referencia. Esta función es la *Normalized Temperature-scaled Cross Entropy* o NT-Xent definida por:

$$\mathcal{L}_\theta^{NT-Xent}(\vec{z}^a, \vec{z}^+, Z^-) = -\ell og\left(\frac{exp(sim(\vec{z}^a, \vec{z}^+)/\tau)}{\sum_{\vec{z_i}^- \in Z^-} exp(sim(\vec{z}^a, \vec{z_i}^-)/\tau))}\right) \tag{3-1}$$

donde *exp* es la función exponencial, τ es un hiperparámetro, llamado de temperatura, Z^- son las proyecciones del conjunto de representaciones R^- definido en el punto 3 y *sim* es la función de similitud coseno dada por:

$$sim(\vec{x}, \vec{y}) = \frac{\langle \vec{x}^\intercal | \vec{y} \rangle}{\tau ||\vec{x}||_2 ||\vec{y}||_2} \tag{3-2}$$

donde $\langle \cdot | \cdot \rangle$ denota el producto interno y $|| \cdot ||_2$ es la norma euclídea.

En general, el proceso de entrenamiento de SimCLR puede dividirse en tres etapas; la primera de ellas consiste en lo que los autores denominan preentrenamiento en el cual se usa el marco de trabajo autosupervisado. La segunda etapa consiste en la transferencia de conocimiento, ya sea como inicialización o extractor de características. La última étapa, es también una transferencia de conocimiento conocida como destilación de conocimiento. El método de SimCLR explora úinicamente las primeras dos etapas, mientras que su versión mejorada incluye las tres.

3.1.2. Entrenamiento autosupervisado o preentrenamiento

Debido a que el marco de trabajo anteriormente expuesto puede llegar a ser confuso, en esta sección se dará una descripción menos formal pero más pictórica del proceso de preentrenamiento en SimCLR y SimCLRv2 (es el mismo para ambos).

Considérese un conjunto de N imágenes no etiquetadas y sea, por simplicidad, el tamaño de lote de dos. Sea TA el conjunto de transformaciones tal y como se define en el marco de trabajo general, puesto a que TA debe aplicarse dos veces sobre cada imagen, en el lote, para obtener el par positivo y de referencia, en este tamaño de lote, se obtienen un total de cuatro imágenes, este proceso se puede observar en la figura 3-1.

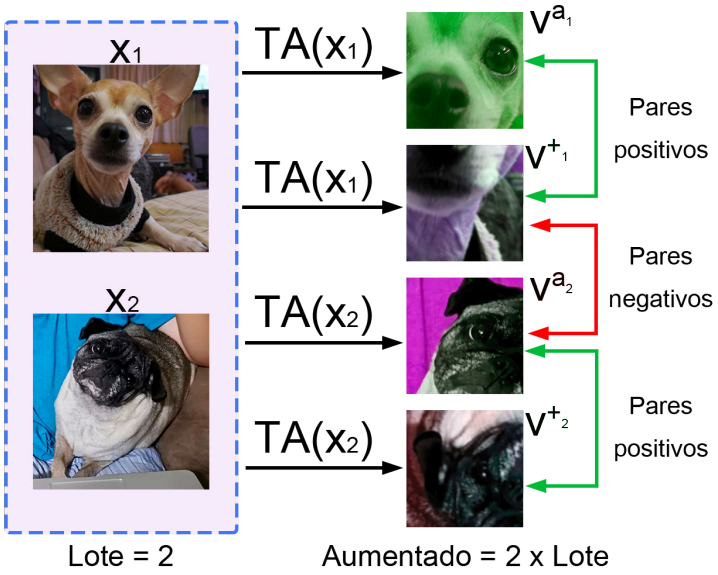


Figura 3-1: Proceso de aumento de datos en SimCLR, con un tamaño de lote de 2. Las imágenes procesadas cumplen las funciones de muestras positivas y/o negativas de acuerdo a cada par. Por ejemplo, el par (v^{a_1}, v^{a_2}) es negativo, al igual que su permutación.

Una vez generadas las imágenes del aumento de datos a través de las imágenes contenidas en un lote, se obtienen las representaciones $\vec{r}$, al ser procesadas por el codificador Resnet-Ω. Posteriormente, cada una de las representaciones obtenidas son proyectadas a otro espacio vectorial a través de un perceptrón multicapa, de esta forma se obtienen las proyecciones $\vec{z}$ de las representaciones, para las cuales se busca maximizar la similitud entre los pares positivos (ver Fig. 3-2)

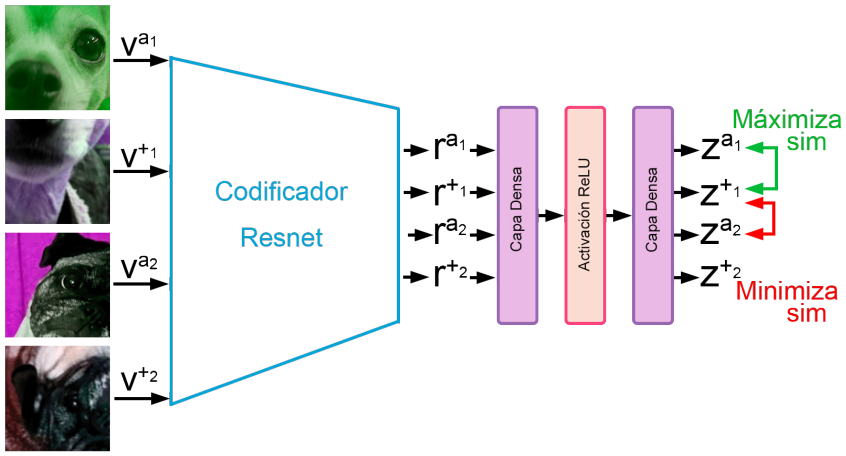


Figura 3-2: Obtención de las representaciones $\vec{r}$ de cada imagen aumentada a través del codificador Resnet-Ω. La representación de cada imagen es, posteriormente, proyectada sobre otro espacio vectorial, para así obtener los vectores $\vec{z}$ correspondientes a cada imagen. Sobre dichas proyecciones se minimiza la función de pérdida NT-Xent.

El cálculo de la similitud entre las proyecciones, así como la función de pérdida tienen una interpretación muy intuitiva. Por un lado, dado que el producto interno no es más que la proyección geométrica de uno de los

vectores sobre el otro, entonces el valor de la similitud coseno es máximo cuando el ángulo entre ambos vectores es cero, y será mínimo cuando el ángulo entre éstos sea $\pi/2$ o $3\pi/2$ (ver figura 3-3), note que el denominador de esta función actúa como un factor de escala.

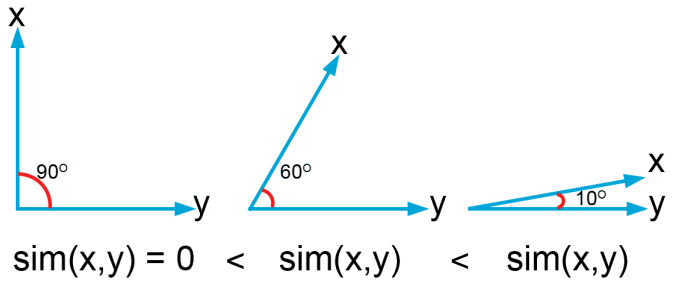


Figura 3-3: Similitud coseno de tres casos diferentes entre los vectores $\vec{x}$ e $\vec{y}$.

Al calcular la similitud entre cada permutación disponible (16 para el ejemplo), es posible obtener un mapa de calor entre las similitudes (ver fig. 3-4). Este mapa de calor no requiere calcularse, sin embargo, es únicamente como fines ilustrativos, pues permite reforzar la intuición de la similitud coseno.

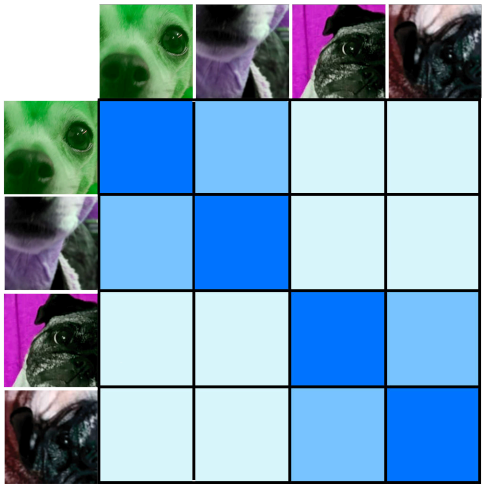


Figura 3-4: Mapa de calor de la similitud de coseno entre pares $(\vec{z}_i, \vec{z}_k)$. Para fines ilustrativos se muestran las imágenes en lugar de sus proyecciones $\vec{z}$.

Nótese que el argumento de la verosimilitud logarítmica definida en la ecuación 3-1 es una función *softmax* aplicada a la similitud entre pares, la cual es equivalente a calcular la probabilidad de que una imagen dada sea similar con respecto a otra imagen de comparación. La figura 3-5 muestra esta idea para el ejemplo de dos imágenes por lote.

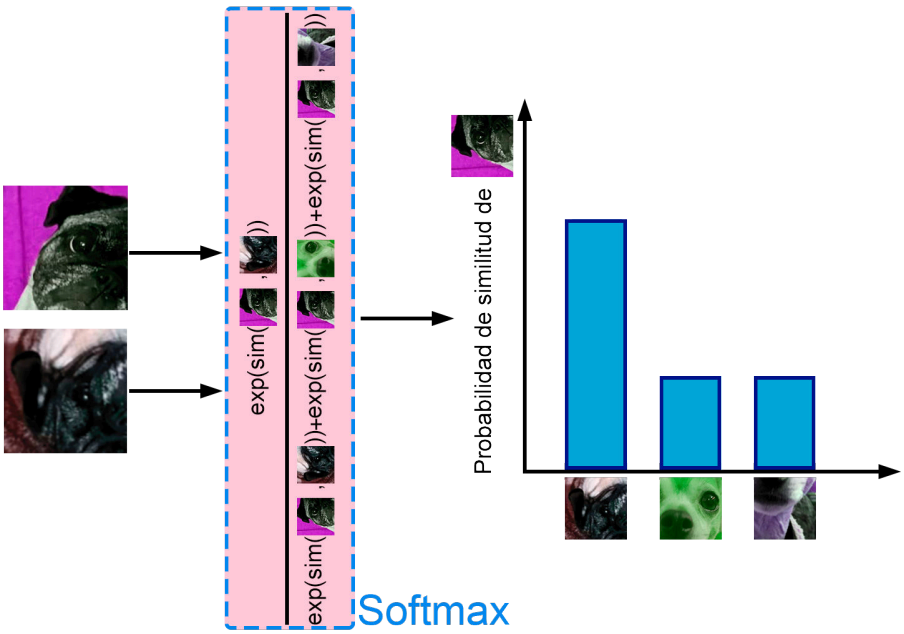


Figura 3-5: Probabilidad de similitud entre una imagen con distintos pares. Para fines ilustrativos se emplean imágenes en lugar de sus proyecciones $\vec{z}$.

Por definición de la función softmax tenemos que:

$$\mathcal{L}_{\theta}^{NT-Xent}(\vec{z}^{a}, \vec{z}^{+}, Z^{-}) \neq \mathcal{L}_{\theta}^{NT-Xent}(\vec{z}^{+}, \vec{z}^{a}, Z^{-}) \tag{3-3}$$

por lo que se define una función de pérdida "total" en la que se toma en cuenta la pérdida de cada par positivo contenido en cada lote. A esta función la denotamos con la letra $\mathcal{L}$ y está definida como:

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} \left[\mathcal{L}_{\theta}^{NT-Xent}(\vec{z}^{a}, \vec{z}^{+}, Z^{-}), \mathcal{L}_{\theta}^{NT-Xent}(\vec{z}^{+}, \vec{z}^{a}, Z^{-}) \right] \tag{3-4}$$

donde N es la cantidad de imágenes en cada lote.
Entonces, la función objetivo a minimizar será 3-4, que permite separar, espacialmente, aquellas proyecciones disimilares entre sí y unir aquellas que sean similares entre sí.

3.1.3. Ajuste fino / entrenamiento semisupervisado

Posterior al preentrenamiento de la red, existen, al menos, dos alternativas para realizar el ajuste fino sobre la misma. La primera de ellas, empleada en SimCLRv2, consiste en tomar la arquitectura Resnet-Ω junto con una cantidad determinada de capas del perceptrón multicapa empleado para la proyección de las representaciones. Mientras que la segunda, consiste en tomar únicamente la arquitectura Resnet-Ω, esta alternativa se aplicó con SimCLR.

Para cualquiera de los dos casos, los autores estudian el rendimiento de la red al realizar ajuste fino o entrenamiento semisupervisado con el 1 %, 10 % y 100 % de datos etiquetados, donde este porcentaje se encuentra contenido en las imágenes utilizadas durante la fase autosupervisada.
Dado que para esta fase se cuenta con un conjunto etiquetado, el ajuste fino consiste en minimizar la función de pérdida *Entropía cruzada categórica*, definida por:

$$CE = -\sum_{i}^{C\ell} t_i \ell og \left(\frac{exp(s_i)}{\sum_j^{C\ell} exp(s_j)} \right) = -\ell og \left(\frac{exp(s_p)}{\sum_j^{C\ell} exp(s_j)} \right) \tag{3-5}$$

donde $C\ell$ es la cantidad de clases, t_i e s_i son las etiquetas reales y la puntuación de la red para cada clase $i \in C\ell$ respectivamente. Si suponemos que las etiquetas se encuentran representadas por la codificación *one-hot* se obtiene la última igualdad. Obsérvese que el argumento de la función logaritmo corresponde a la función softmax.

Finalmente, la evaluación de la calidad de las representaciones aprendidas se realiza mediante una evaluación lineal, donde, como ya se mencionó, se usa la red preentrenada como extractor de características y se entrena un clasificador lineal con el 100 % de las etiquetas.

3.1.4. Resultados sobresalientes

Una vez comprendido el funcionamiento de las arquitecturas de SimCLR y SimCLRv2, en los siguientes apartados se presentarán los resultados más importantes obtenidos en sus respectivos artículos de investigación [12, 27].

SimCLR

Como parte del estudio del aprendizaje de representaciones de calidad por el método autosupervisado, se realiza una comparación por pares entre nueve transformaciones al conjunto de datos. Estas transformaciones son: recorte y redimensionado, volteado sobre el eje vertical, escala de grises, fluctuación de color o jitter, rotaciones con valor $\pi/2$, π y $3\pi/2$, extracción de un parche de la imagen (cutout), ruido gaussiano, desenfoque gaussiano y filtrado de sobel. De esta comparación se destaca que la composición de diversas transformaciones es crucial para la obtención de representaciones de calidad y que la combinación de recorte y redimensionado con distorsión de color son transformaciones esenciales. Los autores argumentan que la fluctuación de color es de suma importancia, pues de esta forma se evita el artificio de distinguir a las imágenes basados únicamente en el histograma de color.

Sobre esta misma dirección, se encuentra que los métodos contrastivos requieren y se ven beneficiados considerablemente por un procesamiento o aumento de datos complejo. En contraparte, los métodos totalmente supervisados no mejoran su rendimiento o incluso se ven afectados con un aumento de datos de la misma índole. De acuerdo con estos hallazgos, el conjunto de aumento de datos en ambas arquitecturas consta de las transformaciones descritas en la subsección 3.1.1.

La proyección de las representaciones obtenidas por el codificador es, también, un factor importante para el aprendizaje de representaciones de calidad. Para probar esto, los autores estudiaron el rendimiento de la arquitectura al preentrenarla utilizando proyecciones lineal, no lineal o sin proyección, así como las dimensiones de salida de las mismas. Se observa que la calidad de las representaciones es mucho mejor cuando se emplea una proyección no lineal con dimensión de 128, aunque para este último no se observan mejoras significativas.

El tamaño de lote, así como la cantidad de épocas de entrenamiento, son otras condiciones que determinan la calidad de las representaciones aprendidas. Se observa que, a menor cantidad de épocas tanto como a menor tamaño de lote la brecha entre el mismo modelo con un mayor tamaño de lote es considerablemente grande. Para ejemplificar, la arquitectura Resnet-50 preentrenada con proyección no lineal y el mismo aumento de datos, durante 100 épocas, con un tamaño de lote de 256 obtiene un valor top-1 en la evaluación lineal de aproximadamente $57.5\,\%$, mientras que con un tamaño de lote de 8192 obtiene un valor aproximado de $64.5\,\%$. Esta brecha se ve reducida al aumentar la cantidad de épocas de entrenamiento, lográndose obtener una diferencia, en el mejor de los casos (1000 épocas), de aprox. $1.7\,\%$. Esta diferencia se debe a que entre más grande es el tamaño del lote, mayores serán las muestras negativas, lo cual facilita la convergencia. Lo mismo sucede al incrementar la cantidad de épocas.

SimCLRv2

Este estudio se centra en el análisis empírico del impacto sobre el rendimiento, tanto en clasificación lineal como en los ajustes finos, al incrementar la profundidad y/o la cantidad de parámetros entrenables de la arquitectura. A diferencia de su primera versión, que cuya red más profunda es Resnet-50 ($\times 4$), SimCLRv2 puede emplear arquitecturas Resnet-{50, 101, 152 y 200} ($\times 1, 2, 3$), con o sin *kernels* selectivos.

Uno de los descubrimientos más importantes se encuentra al relacionar el tamaño de la red con la cantidad de etiquetas disponibles para el ajuste fino; entre más profunda es la red y menor es la cantidad de etiquetas disponibles, la mejora relativa en el rendimiento es mucho mayor. Tomando como referencia la arquitectura Resnet-50, cuya cantidad de parámetros es de aprox. 24 millones, ajustada con el 1 % de las etiquetas, se obtiene un valor top-1 de 57.9 %, mientras que al emplear Resnet-152 (58 millones de parámetros) se obtiene un valor de 64.0 %, al considerar ahora Resnet-101 con *kernels* selectivos la cantidad de parámetros aumenta a 65 millones al igual que el rendimiento top-1 con valor de 68.3 %. De esta manera, los autores argumentan que las arquitecturas más profundas son más eficientes en cuanto a las etiquetas.

Como parte de esta expansión de profundidad, se estudia nuevamente, la proyección de las representaciones, a modo de que el perceptrón multicapa consista de dos, tres y cuatro capas, donde el primero de ellos corresponde a SimCLR. De acuerdo con los autores, sí se cuenta con una cantidad considerablemente pequeña de etiquetas (1 % del total de imágenes), realizar un preentrenamiento con, al menos, 3 capas para la proyección de las representaciones y ajustar desde la capa oculta intermedia, puede mejorar el rendimiento de la tarea semisupervisada, además esta mejora es mayor con modelos pequeños. Sin embargo, si se cuenta con el 100 % de las etiquetas, esta relación no se cumple, siendo incluso perjudicial, por lo que en estos casos, se sugiere realizar los ajustes finos excluyendo el MLP, tal y como se realiza en la primera versión.

3.2. Clasificación morfológica de galaxias en el aprendizaje de máquina

Una búsqueda en las principales bases de datos de artículos científicos, como arXiv, Mendeley y Google Scholar, revela que la cantidad de estudios que emplean métodos autosupervisados para el problema de la clasificación morfológica de galaxias es mucho menor comparada con los métodos supervisados. No obstante, la mayoría de los estudios enfocados en los métodos autosupervisados emplean técnicas de agrupamiento o clustering centrándose en el análisis de la calidad de los mismos a través de las propiedades físicas de las galaxias contenidas en cada uno de éstos, tales como la masa, magnitudes fotométricas en diversas longitudes de onda, la distancia o resolución espacial, entre otros [39–44]. A pesar de que estos estudios son sumamente importantes, éstos se encuentran fuera de nuestro alcance e interés, pues no presentan métricas de clasificación y están más relacionados con el área astronómica. Es por esta razón que en la tabla 3-1 se muestran las características y resultados de dos estudios relacionados que si proveen métricas de clasificación. En contraparte, la tabla 3-2 extiende los resultados presentados en la tabla 1-1 mostrada en el capitulo introducción.

Método	Características			Exactitud
	# imágenes	# clases	¿Desbalance?	
VQ-VAE + HC[1] [45]	7429	2	Si	≈ 87 %
VQ-VAE + HC [45]	7429	2	No	≈ 75 %
Variacion de AlexNet [46]	402804	5	Si	70.10 %
Variacion de AlexNet [46]	402804	5	Si	58.74 %
Variacion de AlexNet [46]	402804	5	Si	68.56 %
Variacion de AlexNet [46]	402804	5	Si	68.60 %

Tabla 3-1: Descripción general de dos estudios relacionados con la clasificación morfológica de galaxias a través del aprendizaje autosupervisado.

Método	Características			Exactitud
	# imágenes	# clases	¿Desbalance?	
Variacion de ResNet [47]	28790	5	Si	95.20 %. 58.62 % clase más desbalanceada
daMCOGCNN [48]	4614	3	Si	97.92 %
SVM [49]	11301	2	Si	84.3 %
Resnet-50 [46]	9362	5	Si	74.63 %. 72.00 % clase más desbalanceada

Tabla 3-2: Descripción general algunos estudios relacionados con la clasificación morfológica de galaxias a través del aprendizaje supervisado. Esta tabla extiende la información proporcionada en la tabla 1-1.

Capítulo 4

Propuesta y conjuntos de datos

La principal meta de este trabajo consiste en la investigación y aplicación del estado del arte de la clasificación de imágenes por el método contrastivo auto y semisupervisados en un área relativamente poco explorada en cuanto a técnicas autosupervisadas, que es la clasificación morfológica de galaxias. Si bien es cierto, que los resultados obtenidos por diversos autores son muy prometedores (ver tablas 1-1 y 3-2), recordemos que estos estudios fueron realizados a través de técnicas supervisadas en donde la cantidad de datos disponible es pequeña y cuya calidad es relativamente buena. De esta forma, en dichos estudios no se tratan las dos grandes cuestiones del aprendizaje de máquina aplicada a la astronomía observacional; ¿cómo atacar el problema del desbalance de clases? y ¿de que manera pueden escalarse estos métodos considerando la baja proporción de etiquetas?
La intención de este trabajo está basada en estas dos cuestiones, que a pesar de no responder a ellas de una manera general, da un pequeñísimo paso hacía esta dirección. De las secciones anteriores se vio que la opción más viable para escalar los métodos del aprendizaje profundo a una cantidad considerable de datos son los métodos autosupervisados pues, con éstos es posible aprender representaciones útiles del conjunto de entrenamiento sin necesidad del uso de etiquetas. Además, es posible hacer uso de los conjuntos de datos etiquetados para ajustar las representaciones aprendidas. La razón por la cual se empleó SimCLR es sencilla; dado que esta arquitectura mostró resultados prometedores, superando sustancialmente los métodos discriminativos, y la naturaleza del problema a resolver, se decidió optar por dicho método.

Con respecto a la cuestión del desbalance de clases, se aplicaron algunas técnicas autosupervisadas, con el fin de balancear el conjunto de entrenamiento, bajo la hipótesis de que las transformaciones dadas por el aumento de datos TA, son, implícitamente, un método de sobremuestreo.

Puesto que la intención de este capitulo radica en explicar de manera muy general, la propuesta planteada, así como los conjuntos de datos empleados en todos los experimentos, en el capitulo 5 se explicará con mayor detalle los procedimientos que se llevaron a cabo para abordar dichas cuestiones.

Antes de proceder a describir los conjuntos de datos empleados durante los experimentos, es necesario conocer el esquema de clasificación morfológica de galaxias de De Vaucouleurs, pues es éste en el que se basó para definir las clases en ambos conjuntos, es por ello que en la siguiente sección se da una breve descripción del mismo.

4.1. Esquema de clasificación morfológica de galaxias de De Vaucouleurs

El esquema extendido de clasificación de Hubble, propuesto por De Vaucouleurs o esquema de De Vaucouleurs describe 18 clases numéricas, llamadas tipos-T o *T-types*, las cuales están basadas en la morfología de las

galaxias [50,51]. Este esquema de clasificación extiende el originalmente planteado por Hubble, el cual constaba de las clases elípticas, espirales, irregulares y lenticulares, al realizar una subdivisión de dichas clases dentro de otras que consideran aspectos morfológicos más detallados. La tabla 4-1 muestra las 18 clases numéricas con su respectiva clase "real".

Tipo-T	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
Etapa	cE	E	E^+	$S0^-$	$S0^o$	$S0^+$	$S0a$	Sa	Sab	Sb	Sbc	Sc	Scd	Sd	Sdm	Sm	Im	?

Tabla 4-1: Clasificación morfológica de galaxias propuesta por De Vaucouleurs. Dentro de ésta existen cuatro grandes clases: galaxias elípticas E, lenticulares $S0$, espirales S e irregulares I. La etapa ? en el tipo-T 11 denota galaxias cuya clasificación dentro de este esquema no es claro.

Las subdivisiones propuestas por De Vaucouleurs consisten en separar las familias espirales y lenticulares en subramas no barradas a, barradas b y débilmente barradas ab, adicionalmente, esta subrama puede dividirse aún más al considerar la forma de sus anillos y/o brazos, lo que deriva en cuatro nuevos tipos a, b, c y d. Por otro lado, los superíndices, en la clasificación de De Vaucouleurs, $-$ y $+$ denotan a las galaxias tempranas y tardías respectivamente. Para la clase lenticular, se añade el superíndice o, que identifica a aquellas galaxias con cierta estructura de dicha clase. Por último, la subrama m se refiere a todas las galaxias similares a las nubes de Magallanes. La figura 4-1 es una representación pictórica de la clasificación extendida de Hubble.

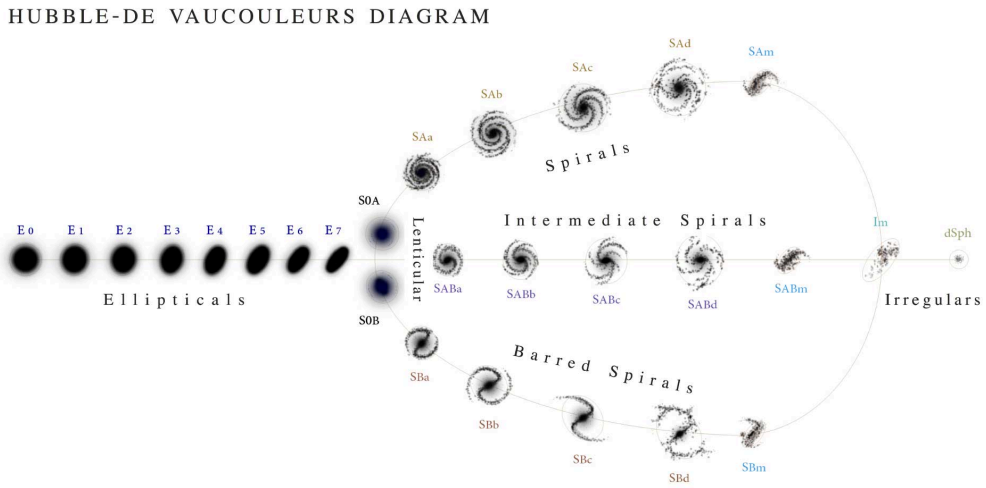


Figura 4-1: Representación pictórica del esquema de clasificación extendido de Hubble propuesto por De Vaucouleurs. Imagen tomada de [52].

Para el desarrollo experimental, se utilizaron dos conjuntos de datos, donde uno de ellos cuenta con etiquetas para 17 clases, mientras que el segundo de ellos no contiene etiquetas, pero es aproximadamente 43 veces más grande que el primero. Las siguientes secciones detallan las características de cada uno de ellos.

4.2. Conjunto etiquetado: Nair

El conjunto de datos etiquetado, que llamaremos Nair, es un subconjunto del conjunto de clasificación ofrecido por Nair y Abraham [53], quienes proveen un catálogo de 14034 imágenes con clasificaciones visuales detalladas. Estas imágenes son, en realidad, un subconjunto del cuarto lanzamiento público de datos del SDSS [54],

en el intervalo de corrimiento hacía el rojo entre el intervalo $(0.01, 0.1)$. Los autores argumentan que la elección de este rango se debe a que los objetos fuera del mismo poseen detalles morfológicos difíciles de detectar. El sistema de clasificación provisto en este estudio está basado en el esquema extendido de clasificación de Hubble, con pequeñas modificaciones en la numeración (ver tabla 4-2) y un total de 14 clases diferentes.

Tipo-T	-5	-5	-5	-3	-2	-2	0	1	2	3	4	5	6	7	8	9	10	99
Etapa	cE	E	E^+	$S0^-$	$S0^o$	$S0^+$	$S0a$	Sa	Sab	Sb	Sbc	Sc	Scd	Sd	Sdm	Sm	Im	?

Tabla 4-2: Esquema de clasificación modificada de De Vaucouleurs presentada en [53].

Para los experimentos propuestos se excluyó la clase 99 debido a su clasificación indefinida, por esta razón, el conjunto Nair consta de 13374 imágenes, divididas en 14 clases. Con el fin de proveer una comparación directa entre el método contrastivo y algunos métodos discriminativos presentados en [46], se redujeron estas catorce clases a una cantidad menor de cinco, tal y como se propone en [46], en donde se redefinen los tipos-T de la tabla 4-2 de la siguiente manera:

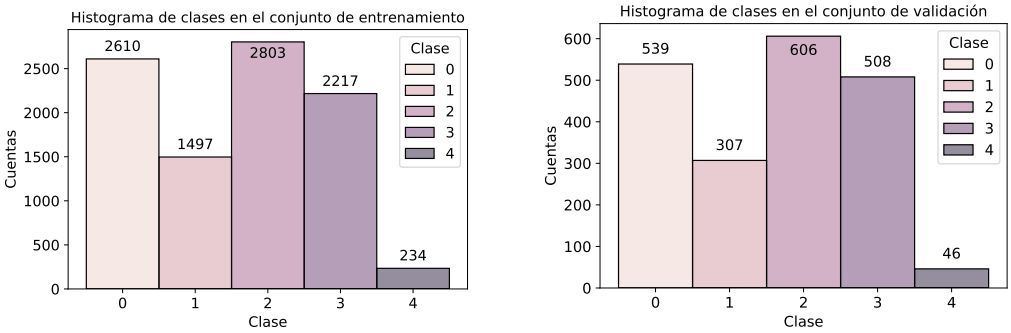
Clase	0	1	2	3	4
Tipos-T	-5, -3	-2, 0	1, 2, 3	4, 5, 6	7, 8, 9, 10
Etapas	cE, E, E^+, $S0^-$	$S0^o$, $S0^+$, $S0a$	Sa, Sab, Sb	Sbc, Sc, Scd	Sd, Sdm, Sm, Im

Tabla 4-3: Definición de cinco clases a partir del agrupamiento de tipos-T, definidos en la tabla 4-2, empleada por [46].

Una vez definidas las clases, el conjunto Nair fue dividido en tres subconjuntos, de entrenamiento, validación y de prueba, procurando mantener la proporción de clases en cada uno de estos. La figura 4-2 muestra los histogramas de clase para las divisiones entrenamiento y validación, mientras que la figura 4-3 muestra la distribución de clases en el subconjunto de prueba.

Como puede observarse, en los tres subconjuntos se tiene un desbalance de clases siendo la número 4 la más desfavorecida. La proporción entre clases, con respecto a la clase mayoritaria, es de $0.93 : 0.53 : 1.00 : 0.79 : 0.08$ para las clases 0, 1, 2, 3 y 4 en el conjunto de entrenamiento, mientras que para los conjuntos de validación y prueba, las proporciones son, respectivamente, $0.88 : 0.51 : 1.00 : 0.83 : 0.08$ y $0.90 : 0.52 : 1.00 : 0.83 : 0.07$.

Por último, la figura 4-4 presenta nueve muestras del conjunto Nair con sus respectivas etiquetas.



(a) Histograma del conjunto de entrenamiento

(b) Histograma del conjunto de validación

Figura 4-2: Distribución de clases en los subconjuntos Nair a) de entrenamiento y b) de validación.

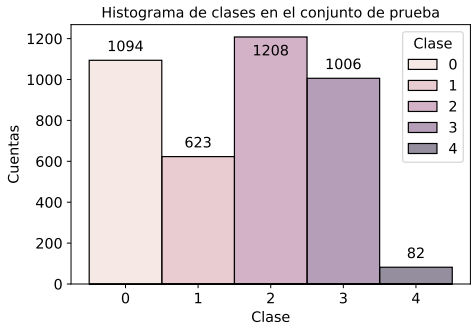


Figura 4-3: Distribución de clases en el conjunto Nair de prueba.

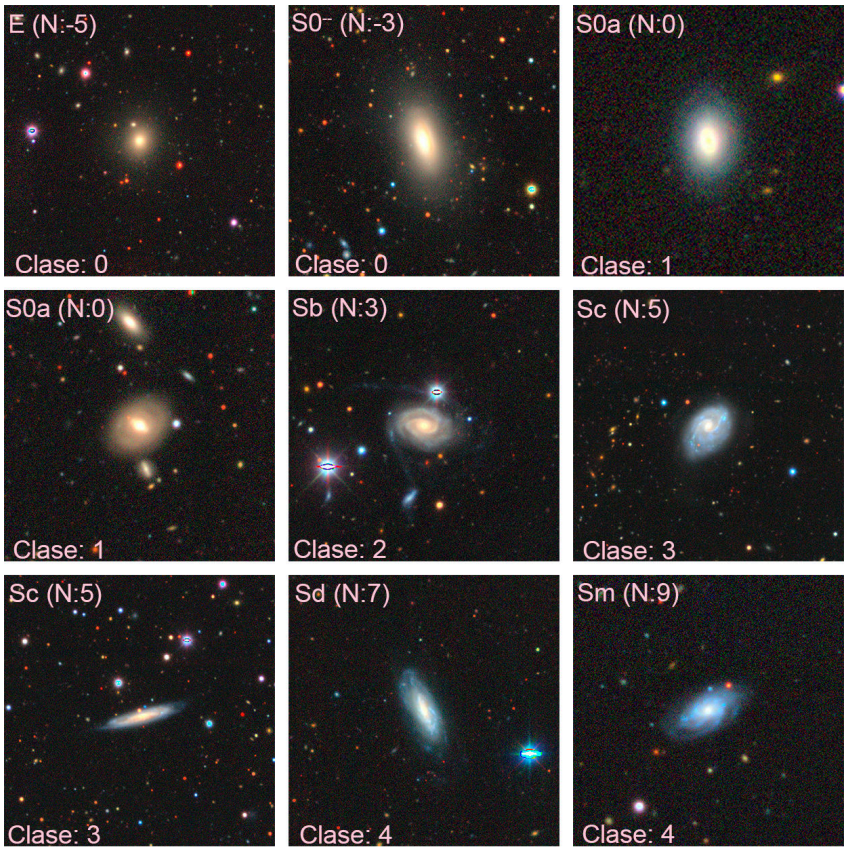


Figura 4-4: Nueve imágenes pertenecientes al conjunto Nair. En la parte izquierda superior se presenta la etiqueta y en paréntesis el tipo-T correspondientes a la tabla 4-2. En la parte inferior izquierda se muestra la clase correspondiente a la tabla 4-3.

4.3. Conjunto no etiquetado: DESI

El catálogo *Nasa-Sloan Atlas* (NSA) [55] es un catálogo de galaxias del SDSS, que cuenta con parámetros de galaxias locales en las bandas ultrtavioleta, óptica e infrarrojo cercano presentes en diversos catálogos como el SDSS, NASA Extragalactic Data Base, ALFALFA, entre otros, siendo el primero de ellos la principal fuente de

información.

Puesto que en el NSA no se proveen imágenes, se usaron algunos de sus parámetros para extraer imágenes provistas por el conjunto DESI. Se destinó un subconjunto del mismo, que consta de 402084 imágenes, para la realización de los experimentos autosupervisados. Dado que, como ya se mencionó, el conjunto DESI contiene galaxias en común con el catalogo de Nair, se aseguró la exclusión de las imágenes en los subconjuntos de prueba y validación de este último.

La figura 4-5 muestra nueve imágenes pertenecientes al conjunto DESI. Cabe mencionar que la adquisición de las imágenes, así como la separación en subconjuntos de entrenamiento, validación y prueba tanto de Nair como DESI se debe al trabajo realizado por [46].

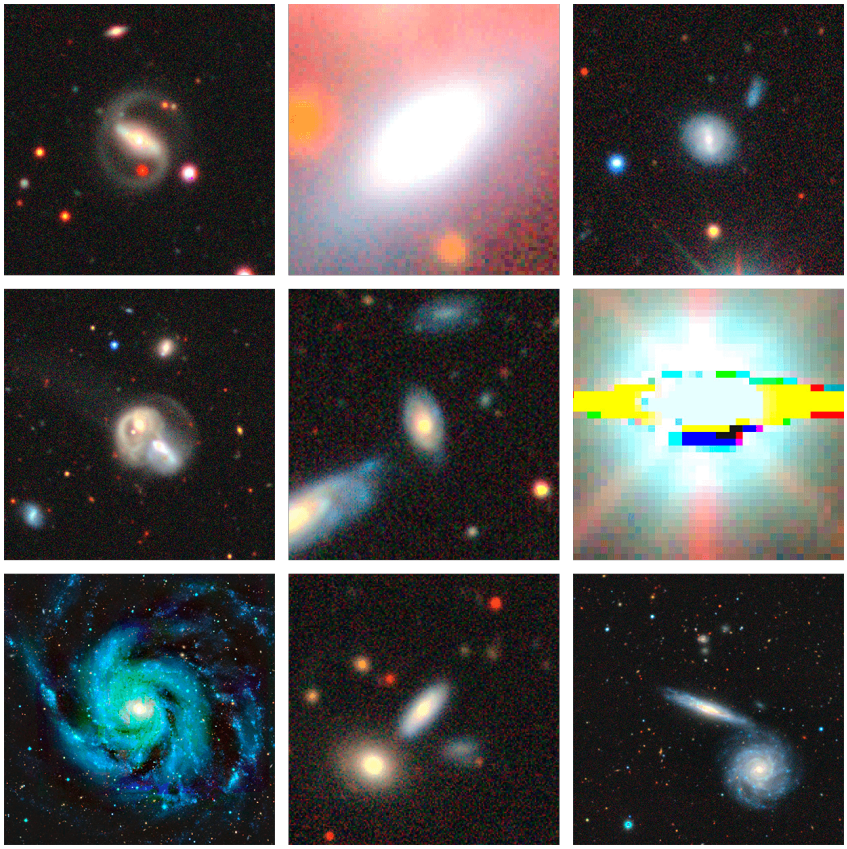


Figura 4-5: Nueve imágenes pertenecientes al conjunto DESI.

Capítulo 5

Desarrollo experimental

En este capítulo se harán explícitos los procedimientos, métodos e ideas generales para comprobar las hipótesis planteadas durante el capítulo de introducción. La primera sección de este apartado se enfocará al primer experimento, que llamamos de referencia, pues será este fundamental para realizar comparaciones entre las modificaciones, mejoras o cambios a alguna de las partes de SimCLR y el resultado provisto por la misma arquitectura sin modificación alguna.

5.1. Experimento de referencia

Puesto que los conjuntos de imágenes ImageNet y DESI son de naturaleza completamente diferente, sería injusto tomar como referencia los valores obtenidos en los artículos de SimCLR [12] y SimCLRv2 [27]. De esta forma, se realizó la primera modificación a SimCLRv2 a modo de que las etapas de preentrenamiento y ajuste fino pudiesen realizarse, únicamente, con el conjunto de datos DESI y Nair respectivamente.

Para este experimento de referencia, no se realizó ninguna otra modificación a la arquitectura; como codificador se usó ResNet-50 sin *kernels* selectivos y un multiplicador $1\times$ en cuanto a su anchura. La proyección de las representaciones provistas por el codificador se realiza a través de un perceptrón multicapa de tres capas, cuya dimensionalidad se ajusta a un valor de 128. Esta arquitectura se preentrenó durante 100 épocas, con un tamaño de lote de 16 imágenes de dimensiones $330 \times 330 \times 3$ píxeles, tasa de aprendizaje de 0.075 con escalado raíz cuadrada y optimizador LARS con momento β de 0.9, hiperparámetro de temperatura τ con valor 0.1; como métodos de regularización se usaron: weight decay de 1×10^{-4}, 10 warmup epochs iniciales y normalización por lotes, siendo los hiperparámetros, los mismos que en SimCLRv2, a excepción de los tamaños de lote e imagen. En cuanto a las transformaciones empleadas durante el preentrenamiento, la tabla 5-1 resume las características de cada una de ellas.

Para la etapa del entrenamiento semisupervisado, se usó la porción de entrenamiento de Nair para dos configuraciones diferentes, la primera de ellas consiste en usar la red preentrenada como extractora de características, entrenando únicamente la capa de clasificación mientras que la segunda configuración consistió en realizar un ajuste fino a toda la red.

Para la primera configuración no podemos considerar que se está realizando una evaluación de las representaciones aprendidas, pues la cantidad de datos etiquetados en función de los no etiquetados es de apenas el $2.33\,\%$, pero sí puede ofrecer una intuición de la calidad de las mismas.

Los parámetros para la primera configuración son los siguientes: 90 épocas con optimizador momentum, tasa de aprendizaje de 0.1, momento de 0.9, tamaño de lote de 64 imágenes con resolución de $330 \times 330 \times 3$ y

Transformación	Probabilidad	Características
Recorte y redimensión aleatorios	1.0	Rango de área a cubrir $[0.08, 1.0]$ Rango relación de aspecto $[\frac{3}{4}, \frac{4}{3}]$ Redimensionado sujeto al tamaño de imagen (en este caso es de 330×330)
Volteado aleatorio	1.0	Solo izquierda - derecha
Distorsión aleatoria de color: brillo, contraste, saturación y hue	0.8	Brillo multiplicativo Contraste con intervalo entre $[0.2, 1.8]$ Saturación con intervalo entre $[0.2, 1.8]$ Hue con máximo de 0.2 Orden de aplicación aleatorio
Escala de grises	0.2	-
Desenfoque gaussiano	0.5	valor σ aleatorio entre $[0.2, 2.0]$

Tabla 5-1: Conjunto de transformaciones aplicadas a cada imagen del lote aumentado durante el preentrenamiento. El orden de aplicación sigue el orden de aparición en la tabla. La probabilidad de aplicación se encuentra en el intervalo $[0.0, 1.0]$.

weight decay de 1×10^{-6}.

Para la segunda configuración, se entrenó la red durante 60 épocas con un tamaño de lote de 64 imágenes con resolución de $330 \times 330 \times 3$, tasa de aprendizaje de 1×10^{-3} con escalado raíz cuadrada y optimizador LARS con momento β de 0.9 y normalización por lotes.

Durante esta etapa semisupervisada, el aumento de datos no es tan fuerte como en el preentrenamiento, pues únicamente se aplican las transformaciones de recorte, redimensión y volteado.

5.2. Experimento de referencia modificado

Con el fin de comparar el experimento de referencia, se desarrolló este experimento, al cual nos referiremos como experimento de referencia modificado, que como su nombre lo indica, se modificó, únicamente, el conjunto de transformaciones empleado durante el entrenamiento autosupervisado, siguiendo las transformaciones correspondientes al experimento 2 o E2 presentado en la sección 5.3, tabla 5-3, pues este resultó ser el más estable. Los hiperparámetros para este experimento son los mismos que para los del experimento de referencia, tanto para el entrenamiento autosupervisado y ajuste fino del clasificador, así como el de toda la red.

5.3. Experimentos transformaciones

Una de las principales hipótesis planteadas se encuentra ampliamente relacionada con el conjunto de transformaciones aplicadas durante la etapa de preentrenamiento autosupervisado. Al realizar una inspección visual en los conjuntos de Nair y DESI, puede determinarse rápidamente que ambos son altamente homogéneos en cuanto a la distribución de color, formas, objetos y alrededores contenidos en cada una de las imágenes. Esta propiedad no es tan significativa cuando se trabaja con el conjunto ImageNet. De esta manera, es factible creer que el artificio de distinguir a las imágenes a través de ciertas características puede acentuarse en este tipo de conjuntos. Para evitar esto, se cree que el conjunto de transformaciones debe reforzarse a modo de que se consideren, también, aspectos relevantes del problema en particular, como por ejemplo, la detección de bordes. En consecuencia, esta reformulación podría, además, mejorar la calidad de las representaciones aprendidas.

Con la intención de probar esta hipótesis se desarrollaron y probaron un conjunto de transformaciones, siguiendo el mismo esquema de preentrenamiento y ajuste fino, aplicado en el experimento de referencia. Por motivos de tiempo y diseño experimental, para esta parte se empleó únicamente el conjunto Nair ajustado, es decir, con la intención de simular un experimento a gran escala, tal y como se hizo en el experimento de referencia, se consideró el conjunto de entrenamiento Nair para el aprendizaje de las representaciones, mientras que para la etapa semisupervisada únicamente se ajustó toda la red a través de solamente el $2.33\,\%$ de imágenes etiquetadas. También, se redujeron las dimensiones de las imágenes a $112 \times 112 \times 3$ píxeles.

Para realizar una comparación justa, se usaron los mismos hiperparámetros tanto en el preentrernamiento, como en el ajuste fino, así como el codificador Resnet-50, para todos los experimentos planteados. Para el método autosupervisado se ajustó una tasa de aprendizaje de 0.075, tamaño de lote de 128 imágenes con dimensiones de $112 \times 112 \times 3$ píxeles, optimizador LARS con momento β de 0.9, hiperparámetro de temperatura $\tau = 0.1$, con métodos de regularización normalización por lotes, weight decay de 1×10^{-4} y cinco warmup epochs iniciales. La proyección de las representaciones se realiza con un perceptrón de tres capas con dimensiones de 128.

Puesto a que la cantidad de imágenes empleadas durante el ajuste fino es una cantidad pequeña, para esta parte se analizó el rendimiento de la red en los conjuntos, sin modificar, de validación y prueba de Nair para cuatro diferentes tasas de aprendizaje, a saber, 5×10^{-3}, 2×10^{-3}, 1×10^{-3} y 7×10^{-4}. Durante esta etapa, se guarda el modelo para aquella época en la que la exactitud es mayor en el conjunto de validación. Al igual que en el preentrenamiento se usó un tamaño de lote de 128 imágenes con dimensiones de $112 \times 112 \times 3$, optimizador LARS con momento de 0.9 y normalización por lotes.

Dado que los experimentos realizados son una combinación de transformaciones base, introduciremos éstas últimas, en forma de tabla (ver tabla 5-2), con su respectiva ”notación” con el fin de facilitar la descripción de cada uno de los experimentos.

La tabla 5-3 muestra las características de cada uno de los experimentos planteados. Con la finalidad de mantener la naturaleza estocástica de las transformaciones, los hiperparámetros introducidos por las transformaciones base se ajustaron a valores aleatorios con una distribución uniforme sobre los rangos admitidos. Al igual que las transformaciones originales, cada transformación se aplica de forma aleatoria con probabilidad p.

5.4. Experimentos K-medias

Una de las hipótesis planteadas al inicio de este trabajo consiste en el estudio de un método de exploración-explotación mediante el cual, sería posible solucionar y/o coadyudar al balance de clases durante la fase de preentrenamiento, esto aún cuando no se cuente con las etiquetas reales del conjunto de preentrenamiento. Para comprender la naturaleza de esta hipótesis, considérese que, en esencia, el método autosupervisado propuesto por SimCLR, permite la separación espacial de la proyección de las representaciones de cada una de las imágenes a través de la pérdida contrastiva definida en el capitulo 3, ecuación 3-4. Al suponer que la distancia entre imágenes disimilares entre sí es lo suficientemente grande, es viable pensar que, el espacio vectorial que contiene dichas representaciones, puede ser dividido en regiones específicas delimitadas por aquellas representaciones similares entre sí, es decir, el espacio vectorial contiene grupos o cúmulos de vectores, donde cada uno de éstos contiene aquellas representaciones similares entre sí. De esta forma, puesto que las representaciones contienen la información de cada imagen, y cada una de ellas corresponden a, al menos, una clase de interés, surge aquí la hipótesis de que la cantidad de cúmulos en el espacio de las representaciones corresponderá a la cantidad de clases reales o de interés. Si bien es cierto que durante las primeras etapas de entrenamiento la cantidad de grupos puede ser incierta e inestable, se espera, que éstos se conformen cada vez más a lo largo de cada época.

Para balancear el conjunto de preentrenamiento se hace uso de esta hipótesis y uno de los algoritmos aglo-

Transformación	Notación	Breve descripción
Invertir imagen	I	Invierte los píxeles de una imagen con la operación $1.0 - imagen$. Sólo se consideran valores de píxel en el rango [1.0, 0.0]
Solarizar	So	Invierte los píxeles de una imagen por encima de un umbral dado
Solarizar e invertir	$So + I$	Solariza una imagen para posteriormente invertirla
Mezcla de Imágenes a, b	M	Calcula la diferencia proporcional d entre b y a, es decir, $d = (b-a)*f$ con f un hiperparámetro entre 0 y 1, la imagen mezclada M será $M = a + d$
Nitidez	N	Modifica la magnitud de la nitidez de una imagen a través de la aplicación de un *kernel* de desenfoque de dimensiones de 3×3, la magnitud está controlada por la transformación M, entre la imagen desenfocada y la imagen original.
Posterizar	P	Reduce el número de bits en cada canal de color de una imagen
Nitidez y posterizar	$N + P$	Modifica la magnitud de nitidez en una imagen para posteriormente posterizarla.
Equalizar	E	Iguala el histograma por canal de una imagen
Oscurecer	D	Sustrae una cantidad determinada a los píxeles de una imagen, es decir: $D = imagen - cantidad$, la cantidad debe estar en el intervalo $(0, 1)$
Autocontraste	AC	Normaliza el contraste de una imagen reajustando el histograma de manera que el píxel más intenso se convierta en 1.0 y el más oscuro en 0.0
Magnitud filtros Sobel	S	Obtiene la magnitud de los filtros sobel en dirección x, S_x e y, S_y, es decir, $S = \sqrt{S_x^2 + S_y^2}$, previo a la obtención de S_x y S_y a la imagen se le aplica desenfoque gausiano con valor σ_x , σ_y enteros aleatorios entre 3 y 13
Recorte central	CC	Recorta una imagen al centro según un valor de proporción entre $(0, 1]$, en caso de que dicho valor sea uno, la imagen no se modifica.

Tabla 5-2: Conjunto de transformaciones base. Los experimentos planteados son una mezcla de éstas.

merativos no supervisados más populares, que es K-medias, el cual permite definir anticipadamente la cantidad de grupos, sobre el cual estará definido este algoritmo. La metodología general para coadyudar al balance de clases durante el entrenamiento autosupervisado es como se describe a continuación:
Durante una época de entrenamiento, se guarda una copia de cada vector de representación (salida del codificador) de cada una de las imágenes contenida en el lote. Una vez finalizada la extracción de características de cada imagen contenida en el conjunto de entrenamiento, se ejecuta el algoritmo de K-medias sobre las representaciones almacenadas y se procede a extraer una pseudoetiqueta para cada uno de estos vectores, dicha etiqueta corresponde al número o asignación de grupo provisto por el algoritmo. Una vez conocidas estas etiquetas, es posible separar las imágenes (nombres) con su respectiva pseudoetiqueta, permitiendo redefinir el conjunto de datos al considerar un balance automático a través de dichas pseudoetiquetas. Este nuevo conjunto de datos servirá como base de la siguiente época.

Como puede observarse, este proceso es iterativo y cambiante a lo largo de cada época de entrenamiento. Cabe mencionar que para evitar sesgos en las pseudoetiquetas, el algoritmo de K-medias, es reiniciado por cada época y éste es ejecutado independientemente de la pérdida y actualización de pesos del algoritmo de aprendizaje.

A pesar de no considerar directamente los resultados provistos por el algoritmo aglomerativo sobre la función de pérdida y/o actualización de pesos, éstos sí pueden ser considerados e incluidos de una manera relativamente sencilla, sin embargo, esto no fue aplicado pues, debido a la naturaleza estocástica de la asignación de grupos, la inclusión de una función de pérdida relacionada a la predicción de las pseudoetiquetas podría ser muy problemática, ya que al diferir éstas a lo largo de cada época, la predicción de las mismas estaría ajustándose a valores aleatorios en el intervalo $[0, clases - 1]$, en consecuencia dicha función podría ser altamente inestable. Para

Experimento	Transformación (probabilidad p)							Otros
	D	$N+P$	E	$So+I$	S	CC	O	
E1	✓(0.5)	✓(0.3)	×	✓(0.2)	✓(0.35)	✓(1.0)	✓	Recorte central a proporción 0.7 Recorte y redimensión con rango de área a cubrir de $[0.3, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E2	✓(0.5)	×	×	×	✓(0.3)	✓(1.0)	✓	Recorte central a proporción 0.65 Recorte y redimensión con rango de área a cubrir de $[0.2, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E3	✓(0.5)	×	×	✓(0.2)	✓(0.3)	✓(1.0)	✓	Recorte central a proporción 0.65 Recorte y redimensión con rango de área a cubrir de $[0.2, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E4	×	×	×	×	×	×	✓	Transformaciones originales, con las mismas características que en la tabla 5-1 excepto los intervalos de contraste y saturación ajustados $[0.6, 1.4]$ y Hue con máximo de 0.1
E5	×	×	×	×	×	✓(1.0)	✓	Recorte central a proporción 0.65 Recorte y redimensión con rango de área a cubrir de $[0.2, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E6	✓(0.5)	✓(0.5)	×	✓(0.2)	✓(0.3)	✓(1.0)	✓	Recorte central a proporción 0.65 Recorte y redimensión con rango de área a cubrir de $[0.2, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E7	✓(0.5)	×	×	✓(0.3)	✓(0.45)	×	✓	Recorte y redimensión con rango de área a cubrir de $[0.08, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E8	×	×	×	×	×	✓(1.0)	✓	Recorte central a proporción 0.7 Recorte y redimensión con rango de área a cubrir de $[0.3, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E9	✓(0.1)	×	×	✓(0.2)	✓(0.35)	✓(1.0)	✓	Recorte central aleatorio en el intervalo $[0.8, 0.65]$ Recorte y redimensión con rango de área a cubrir de $[0.1, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1 Desenfoque gaussiano para el filtro de sobel con valor entero sigma aleatorio $\sigma_x = \sigma_y \in [3, 7]$
E10	✓(0.1)	×	×	✓(0.2)	✓(0.35)	✓(1.0)	✓	Recorte central aleatorio en el intervalo $[0.72, 0.65]$ Recorte y redimensión con rango de área a cubrir de $[0.18, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E11	×	×	×	✓(0.25)	✓(0.4)	✓(1.0)	✓	Recorte central aleatorio en el intervalo $[0.72, 0.65]$ Recorte y redimensión con rango de área a cubrir de $[0.18, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1
E12	✓(0.5)	×	×	✓(0.2)	✓(0.3)	✓(1.0)	✓	Recorte central aleatorio en el intervalo $[0.75, 0.55]$ Recorte y redimensión con rango de área a cubrir de $[0.2, 1.0]$ Intervalo de contraste y saturación de $[0.6, 1.4]$, Hue con máximo de 0.1

Tabla 5-3: Experimentos propuestos con sus características. O denota las transformaciones originales planteadas en la tabla 5-1, cuyas modificaciones se aclaran en la columna *Otros*.

superar esta barrera, debe considerarse un método de asignación más robusto, que evite la variabilidad de las pseudoetiquetas o implementar el método propuesto por [56], el cual se divide en dos fases: aglomerado a través de K-medias, para posteriormente realizar un aprendizaje de las características basado en las pseudoetiquetas provistas en la primera fase. Esta última metodología queda fuera del alcance de SimCLR, sin embargo podría, también ser implementado.

Para estudiar el comportamiento del método aquí propuesto, se realizaron tres diferentes configuraciones: 1) preentrenamiento autosupervisado ajustando en cada época el conjunto de entrenamiento a través de K-medias por 100 épocas, 2) preentrenamiento autosupervisado ajustando en las primeras 50 épocas el conjunto de entrenamiento a través de K-medias por 100 épocas, donde las últimas 50 épocas se realizan con el conjunto de entrenamiento de Nair y 3) preentrenamiento autosupervisado ajustando en las últimas 50 épocas el conjunto de entrenamiento a través de K-medias por 100 épocas, donde las primeras 50 épocas se realizan con el conjunto de entrenamiento de Nair. Por motivos de estabilidad, en el entrenamiento autosupervisado, se emplearon las transformaciones del experimento 2 de la tabla 5-3, pues fue este último el cual presentó una variación mínima con respecto a los valores de exactitud en el conjunto de prueba para todas las tasas de aprendizaje empleadas. Los hiperparámetros tanto de preentrenamiento, como de ajuste fino fueron los mismos que los empleados para la realización de los experimentos de transformaciones y se usó, para el ajuste fino, el mismo subconjunto que consta del 2.33 % de las etiquetas. Esto nos permitirá realizar una comparación justa entre los experimentos que

no consideran K-medias como un método de balanceo y los planteados en esta sección.

Finalmente, cabe mencionar que se probaron diferentes algoritmos de agrupamiento, tales como agrupamiento jerárquico, Gaussian Mixture y agrupamiento espectral. Sin embargo, éstos fueron descartados debido a problemas en la convergencia y/o tiempo de convergencia.

5.5. Experimentos supervisados y efectividad del método autosupervisado

Hasta ahora, todos los experimentos realizados se han enfocado en realizar comparaciones directas del rendimiento y/o comportamiento entre el método original y los propuestos durante las secciones anteriores, las cuales son variaciones del método original. En esta sección, el enfoque cambiará a diversas configuraciones que permitan realizar comparaciones directas entre los métodos supervisados y semisupervisados. A pesar de que ambos pueden no estar muy correlacionados entre sí, esta comparación da la facultad de discernir las fortalezas y debilidades del método autosupervisado, así como conocer la dificultad del problema al cual se enfrenta, aún con el uso de la totalidad de las etiquetas.

Para estos fines se proponen los siguientes experimentos:

- *Método completamente supervisado desde cero*: Este experimento sigue la misma arquitectura y función de costo que en el ajuste fino presentado con anterioridad. La única diferencia entre este experimento y uno de ajuste fino, consiste en la inicialización de pesos; que para este caso será aleatorio y, además, la tasa de aprendizaje es aumentada a un valor de 0.3 con escalado lineal, que de acuerdo a los autores de SimCLR, este último es equivalente a un valor 0.075 con escalado raíz cuadrada.
- *Método completamente supervisado con pesos de ImageNet*: Como su nombre lo indica, durante este experimento se inicializaron los pesos del codificador ResNet con los pesos aprendidos para la tarea de clasificación en ImageNet. De este método se considerará, solamente, un experimento:
    1. Ajuste fino: Los pesos de toda la red se ajustan a la tarea de clasificación morfológica de cinco tipos de galaxias. Para este caso empleamos los mismos hiperparámetros que en un ajuste fino de los experimentos anteriores.

Para todos los casos se usó un tamaño de imagen de $330 \times 330 \times 3$ píxeles y, con el fin de analizar el impacto del desbalance de clases, se entrenó cada arquitectura, por separado, con un conjunto de validación y entrenamiento balanceados y desbalanceados.

Por otro lado, para probar la efectividad del método autosupervisado, es decir, para demostrar que el aprendizaje de las representaciones durante el preentrenamiento aporta información relevante para la tarea de clasificación, así como para el rendimiento de la misma, se realizó un experimento de control en donde se realiza un ajuste fino, bajo las mismas condiciones que las descritas anteriormente, a una red cuya inicialización fue aleatoria.

5.6. Experimentos con otros codificadores

Se cree que el cambio de codificador por alguno más actual puede revelar hechos que colaboren en la adquisición de una intuición mayor de los métodos auto y semisupervisados, pues la función principal del codificador consiste en la extracción de las características visuales del conjunto de datos de interés. Por lo que, al cambiar de codificador, se espera un cambio interno en dicha extracción. Sin embargo, este cambio interno no es sencillo de comprender e interpretar.

Para diseñar este experimento, se planteó la posibilidad de usar la red AlexNet empleada por [46], pero ésta fue descartada debido a su baja complejidad con respecto al codificador original. Puesto a que las redes profundas han mostrado un poder expresivo mucho mayor comparado con su contraparte menos profundas y siguiendo la idea de que aquellas arquitecturas novedosas muestran una mejoría significativa en cuanto al rendimiento y/o eficiencia computacional, se seleccionaron dos codificadores: DenseNet-161 [57], ya que sigue una arquitectura residual similar a la de la red ResNet-Ω y EfficientNetV2 de tamaño medio [58]. Dado que nuestro único interés radica en la comparación cuantitativa entre estas dos arquitecturas y la arquitectura original, no se presentarán los detalles técnicos de ambas arquitecturas.

Con el fin de realizar una comparación justa, se optó por emplear los mismos hiperparámetros tanto para el entrenamiento autosupervisado y ajuste fino, así como los mismos conjuntos de datos que los empleados durante la sección Experimentos Transformaciones (5.3). En cuanto al procesamiento de datos durante el preentrenamiento, se usaron las transformaciones empleadas por el experimento E2 (transformaciones originales con recorte central a proporción de 0.65).

5.7. Visualización

Los resultados obtenidos durante los experimentos anteriormente planteados son meramente cuantitativos y fueron obtenidos a través de técnicas ampliamente conocidas. Durante esta sección se presentarán resultados cualitativos, los cuales pueden dar pistas intuitivas acerca del funcionamiento, procesamiento y estructura general de las arquitecturas consideradas más relevantes. Para estos fines se usó el algoritmo *SHapley Additive exPlanations* (SHAP) para visualizar las regiones de decisión de la red y se obtuvieron algunas cuadriculas de activación para determinadas capas de cada modelo, entre otros artefactos de visualización propuestos por los algoritmos Lucid y Lucent.

5.7.1. SHapley Additive exPlanations

SHAP es un algoritmo planteado en [59] descrito como un marco de trabajo unificado para la interpretación de las predicciones producidas por diversos algoritmos de inteligencia artificial, aprendizaje de maquina así como de aprendizaje profundo a través de la asignación de valores medibles de importancia a cada característica para una predicción en particular. De acuerdo con sus autores, la importancia del algoritmo SHAP radica en la identificación de un conjunto medible de características aditivas relacionado a su importancia y la existencia de una solución única en dicho conjunto.

Puesto a que el interés de este trabajo radica en explicar un conjunto de imágenes a través del algoritmo SHAP, se expondrá a continuación una sencilla explicación de su funcionamiento en dichos objetos, así como una imagen de ejemplo.

[59] *La imagen 5-1 explica diez salidas (números del 0 al 9) para cuatro imágenes diferentes. Los píxeles rojos aumentan la salida del modelo, mientras que los azules disminuyen. Las imágenes de entrada se muestran a la izquierda, y como fondos en escala de grises detrás de cada una de las explicaciones o salidas. La suma de los valores SHAP es igual a la diferencia entre el resultado esperado del modelo (promediado sobre el conjunto de imágenes de fondo) y el resultado actual del modelo. Observe que para la imagen del número cero el centro es importante, mientras que para la imagen del número cuatro la falta de conexión en la parte superior hace que la predicción sea un cuatro en lugar de nueve.*

5.7.2. Activaciones fuertes por capas

Una técnica interesante de visualización que permite interpretar algunos patrones aprendidos por uno o un conjunto de filtros de una red neuronal se conoce como visualización por optimización, la cual está basada en el algoritmo del descenso del gradiente [60]. Esta técnica busca responder a la pregunta: Dado un filtro f,

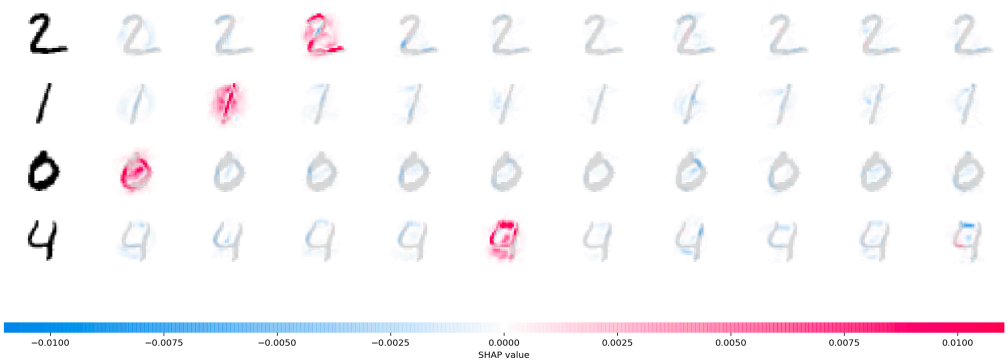


Figura 5-1: Imagen SHAP. Tomada de https://raw.githubusercontent.com/slundberg/shap/master/docs/artwork/mnist_image_plot.png.

correspondiente a una capa determinada L, ¿Qué es lo que dicho filtro f ha aprendido a detectar? o, en otras palabras, ¿cuál es la imagen de entrada tal que maximiza la activación del mapa de características generado por el filtro f? De esta pregunta surgen diversos métodos de optimización, así como características de visualización, de estas últimas, es posible obtener diferentes imágenes de entrada que activan una neurona o filtro y/o realizar un combinación de las mismas.

Nuestro interés particular radica en responder ¿Cómo los diferentes canales dentro de cada capa transforman o visualizan una imagen dada? y ¿Cuáles son las activaciones fuertes para cada capa para una imagen dada? La visualización planteada por la primer pregunta se conoce como cuadriculas de activación o *activation grids*, mientras que la segunda de ellas se conoce como inversión de características. A modo demostrativo se presentan ambas visualizaciones obtenidas por el algoritmo Lucent en el modelo inceptionv1 en las figuras 5-2 y 5-3.

5.8. Resumen experimentos

Los experimentos planteados durante esta sección así como sus objetivos se resumen en la tabla 5-4.

Figura 5-2: (a) imagen de entrada, (b) cuadriculas de activación del modelo inceptionv1 para la capa mixed4d. Imágenes tomadas de Lucent.

Figura 5-3: Inversión de características de la imagen original para diferentes filtros del modelo inceptionv1 (nombre en parte superior). Imágenes tomadas de Lucent.

Experimento	Objetivos
Experimento de referencia	Implementar y reproducir SimCLRv2 para la tarea de clasificación morfológica de galaxias, usando los mismos hiperparámetros propuestos en el artículo de investigación. Obtener métricas de evaluación como la matriz de confusión, curvas ROC-AUC, Precision-Recall, así como los valores de exactitud, recall, F1 para cada clase.
Experimentos transformaciones	Determinar el conjunto de transformaciones más estable y con mejores resultados en cuanto al valor de exactitud, aplicable al conjunto de datos astronómico. Validar, experimentalmente, una posible correlación entre el mejor conjunto de transformaciones y la morfología de las galaxias.
Experimento de referencia modificado	Implementar SimCLRv2 con el mejor conjunto de transformaciones encontrado durante los experimentos de transformaciones, para así, obtener métricas de evaluación como la matriz de confusión, curvas ROC-AUC, Precision-Recall y los valores de precision, recall, F1, con el fin de compararlos directamente con la implementación original.
Experimentos K-medias	Analizar el impacto sobre la detección de verdaderos positivos en todas las clases al implementar SimCLR con K-medias, de modo que, al final de cada época del entrenamiento autosupervisado, pueda balancearse el conjunto de entrenamiento a través de las etiquetas generadas por el algoritmo aglomerativo.
Experimentos supervisados	Implementar una arquitectura similar a SimCLRv2 y entrenarla a través del método completamente supervisado. Realizar ajuste fino de la misma arquitectura entrenada con ImageNet. Comparar los resultados de estas dos redes neuronales con el fin de conocer la complejidad del problema con el que se trata.
Efectividad del método autosupervisado	Demostrar, indirectamente, que el aprendizaje de las representaciones generales, obtenida por SimCLR durante la fase autosupervisada, son útiles durante la fase semisupervisada.
Experimentos con otros codificadores	Explorar posibles mejoras en el rendimiento de SimCLR al sustituir el codificador Resnet-50 con otros dos codificadores más actuales, que han probado superar al codificador original en tareas supervisadas.
Visualización	Explorar cualitativa y visualmente las regiones de decisión, pesos aprendidos y activaciones fuertes de algunos modelos derivados de la exploración cuantitativa. Visualizar la distribución de clases real e inferida por el algoritmo K-medias aplicado sobre las representaciones aprendidas a través de SimCLR, inferir la distribución de clases en el conjunto no etiquetado y determinar la cantidad de clases óptima.

Tabla 5-4: Resumen de los objetivos planteados para cada experimento a realizar.

Capítulo 6

Resultados y discusión

En este capítulo se presentarán los resultados obtenidos durante los experimentos propuestos en el capítulo 5, así como una discusión de los mismos. Por motivos de espacio y legibilidad, únicamente se presentarán figuras, tablas y/o resultados más destacados. Aquellos resultados secundarios, relevantes y no repetitivos, serán presentados en el apéndice A al final de este trabajo. Además, como guía visual se añade la figura 6-1, la cual sintetiza de una forma amigable todos los experimentos realizados y muestra la conexión (comparaciones y/o discusiones) entre cada uno de ellos.

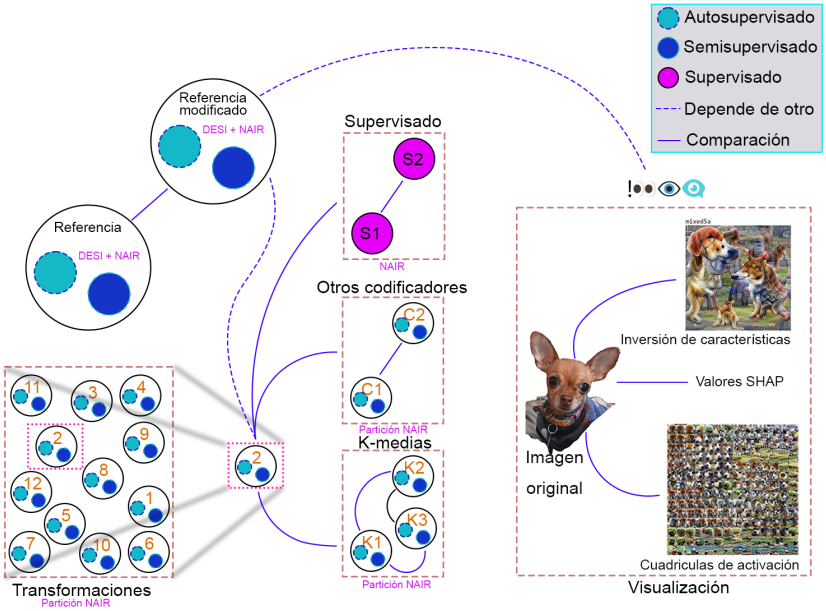
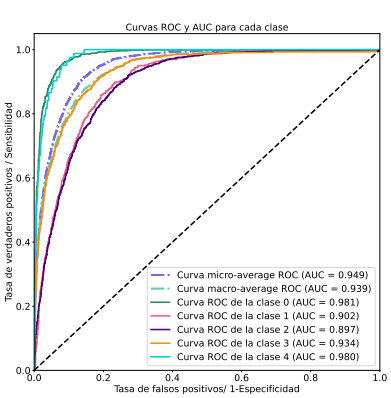


Figura 6-1: Conexión entre todos los experimentos. Una linea continua denota una comparación directa entre resultados. La linea discontinua denota una dependencia, es decir, el experimento de referencia modificado requirió de los resultados obtenidos durante el experimento de transformaciones. Dentro de las transformaciones los números representan cada uno de los 12 experimentos, de aquí se selecciona el considerado más estable (2).

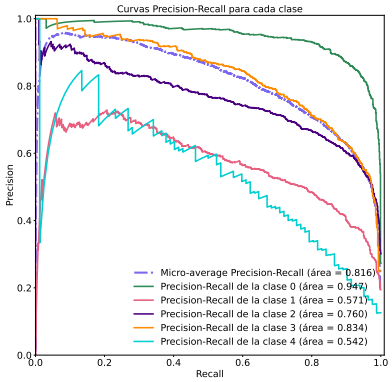
6.1. Experimento de referencia

6.1.1. Codificador como inicialización

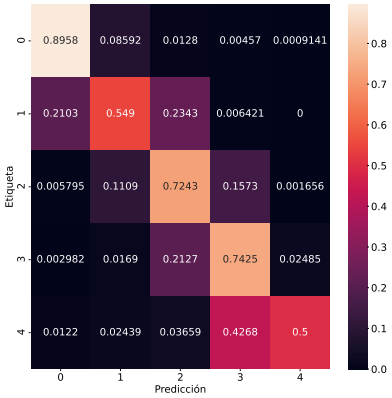
Las métricas de evaluación obtenidas durante el ajuste fino, usando el clasificador como inicialización y los conjuntos de entrenamiento y validación desbalanceados se muestran en la figura 6-2, mientras que la figura 6-3 corresponde al uso de los conjuntos de datos balanceados.



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

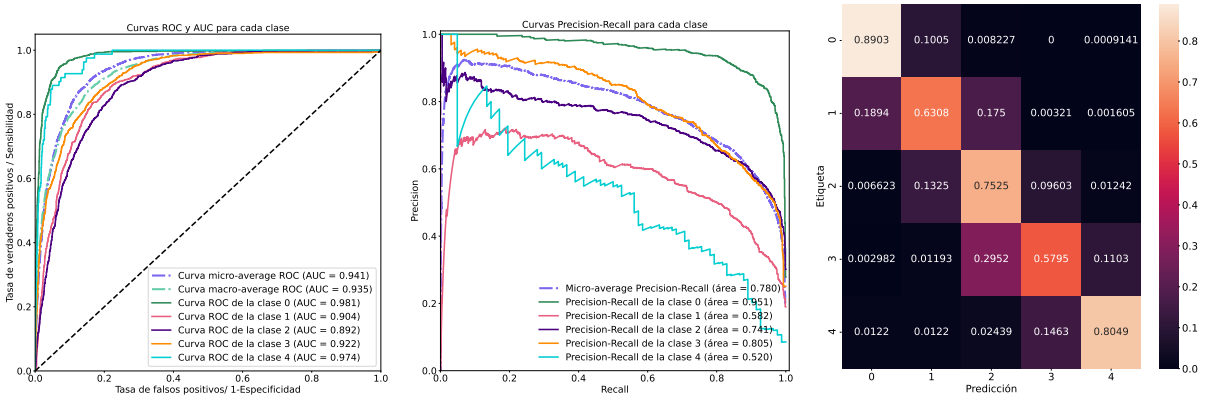
Clase	Precision	Recall	F1
0	87.34 %	89.58 %	88.45 %
1	58.06 %	54.90 %	56.44 %
2	69.89 %	72.43 %	71.14 %
3	76.15 %	74.25 %	75.19 %
4	59.42 %	50.00 %	54.30 %
macro avg	70.17 %	68.23 %	69.10 %
weighted avg	74.17 %	74.83 %	74.25 %
Exactitud	74.83 %		

(d) Reporte de clasificación

Figura 6-2: Métricas de evaluación obtenidas para el experimento de referencia usando el codificador como inicialización y conjuntos de datos desbalanceados.

Como era de esperar, al usar un conjunto balanceado, el modelo es capaz de identificar, a mayor proporción, la cantidad de verdaderos positivos para las clases minoritarias (Clases 4 y 1), con respecto a su contraparte desbalanceada. A pesar de que la cantidad de muestras de la clase 3, es comparable a la cantidad de las dos clases mayoritarias, para esta clase, la correcta detección de verdaderos positivos disminuye 16.3 %. De esta manera, la diferencia, significativa, no puede ser atribuida a una baja cantidad de muestras, pues observemos, también, que el cambio en el valor recall para las clases mayoritarias es insignificativo. Nótese que para los casos balanceado y desbalanceado, la proporción de falsos positivos, en la clase 3, se concentra en su clase contigua 2. Recordando la asignación de clases en la tabla 4-3, nos encontramos una característica común entre las clases 2 y 3: Ambas son, esencialmente, de tipo espiral, cuya diferencia más prominente es la forma, cantidad y separación de los brazos. Ahora bien, la clase cuatro, también contiene un subconjunto de tipos espirales, los cuales se destacan por poseer alguna especie de irregularidad, esto, también, podría justificar el porqué la cantidad de falsos positivos entre estas dos clases no es muy cercana a cero. De esto, es factible atribuir esta disminución en el valor de verdaderos positivos de la clase 3, a una fuerte transición continua entre ésta y su clase contigua 2.

A pesar de que la proporción de falsos positivos se concentra en las clases contiguas para todas las clases, se observa que, al balancear el conjunto de datos, existe una pequeña disminución de dicha proporción, excepto



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	88.22 %	89.03 %	88.62 %
1	58.13 %	63.08 %	60.51 %
2	68.55 %	75.25 %	71.74 %
3	81.77 %	57.95 %	67.83 %
4	34.02 %	80.49 %	47.83 %
macro avg	66.14 %	73.16 %	67.31 %
weighted avg	74.91 %	72.89 %	73.13 %
Exactitud	72.89 %		

(d) Reporte de clasificación

Figura 6-3: Métricas de evaluación obtenidas para el experimento de referencia usando el codificador como inicialización y conjuntos de datos balanceados.

para la clase 3. Por lo que, si se imaginase una recta, que representa la transición continua, la distancia entre cada clase (puntos sobre la recta), estaría directamente relacionada con la proporción de falsos positivos, por ejemplo, de las matrices de confusión, la clase tres se encontraría más cercana a la clase dos con respecto a la clase cuatro.

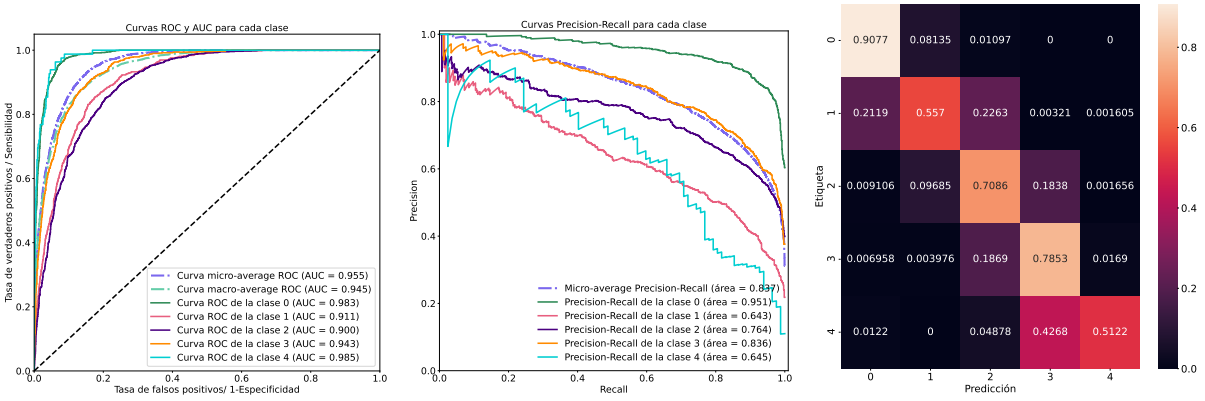
6.2. Experimento de referencia modificado

6.2.1. Codificador como inicialización

Los resultados obtenidos para el experimento de referencia modificado, usando el codificador como inicialización se muestran en la figura 6-4. A diferencia de la figura 6-4, la figura 6-5 muestra las métricas obtenidas al usar los conjuntos de validación y entrenamiento balanceados.

De las figuras 6-4 y A-3, se observa que, el hecho de entrenar la capa de clasificación repercute significativamente sobre las métricas de clasificación, lográndose, obtener hasta una disminución del 56.3 % en el valor del área bajo la curva Precision-Recall para la clase 4, al considerar la curva micro-averange, la diferencia es, también significativa, de 29.4 %. Por otro lado, la diferencia de área bajo las curvas micro-averange ROC-AUC es del 11.9 %. Al observar los valores $F1$, se encuentra una mejoría significativa para las clases 2 y 4 considerando la red usada como inicialización. Estos resultados son totalmente esperados, pues existe una clara ventaja sobre el ajuste fino de la capa de clasificación.

Comparemos ahora, los resultados obtenidos por las redes usadas como inicialización. Se observa que todos



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

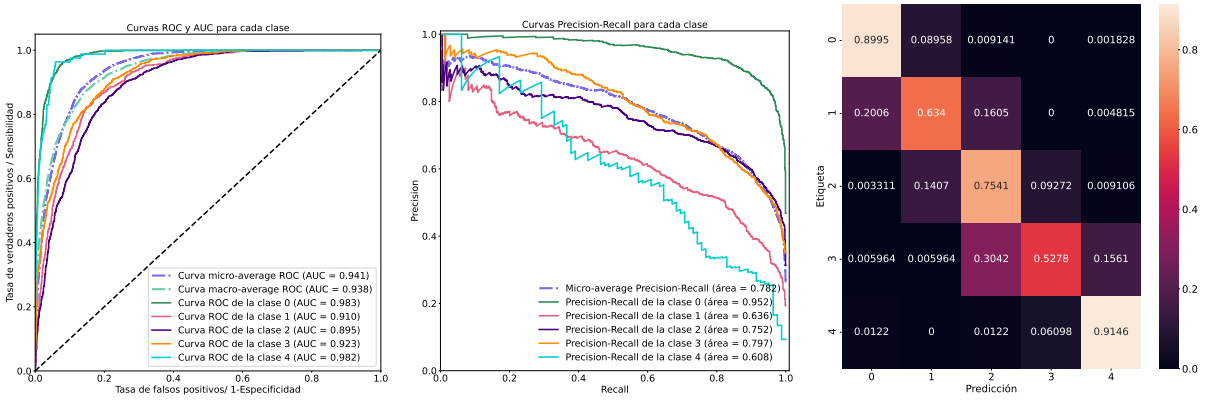
Clase	Precision	Recall	F1
0	86.80 %	90.77 %	88.74 %
1	62.30 %	55.70 %	58.81 %
2	71.27 %	70.86 %	71.07 %
3	75.31 %	78.53 %	76.89 %
4	67.74 %	51.22 %	58.33 %
macro avg	72.68 %	69.42 %	70.77 %
weighted avg	75.05 %	75.46 %	75.18 %
Exactitud	75.46 %		

(d) Reporte de clasificación

Figura 6-4: Métricas de evaluación obtenidas para el experimento de referencia modificado usando el codificador como inicialización y conjuntos de validación y prueba desbalanceados.

los valores del área bajo la curva de las curvas ROC-AUC y Precision-Recall no presentan variaciones significativas, lo mismo sucede para los valores $F1$, en donde, únicamente se observan grandes diferencias en las clases 3 y 4. A pesar de que se esperaba, que la matriz de confusión, para la red entrenada con conjuntos balanceados, presentara valores homogéneos en su diagonal, tal y como sucedió con la figura A-4c, esto no fue así. Lo cual podría ser explicado por el preprocesamiento aleatorio de las imágenes en el ajuste fino, sin embargo, este argumento no parece ser lo suficientemente sólido como para justificar este hecho. Por lo que se desconoce la razón por la cual pudo haber sucedido. Pese a esto, la red mejora significativamente las predicciones correctas para las clases 4, 1, 2 y una pequeña disminución para la clase 0. Al igual que en el experimento de referencia, la clase 3 sufre una importante disminución en la proporción de verdaderos positivos, al emplear los conjuntos de datos balanceados. Lo cual es atribuido, nuevamente, a la transición entre clases.

Al comparar el valor macro average de la métrica recall entre el experimento de referencia y el modificado, se encuentra que existen diferencias a favor de este último de $1.19\,\%$ y $1.49\,\%$ para los modelos entrenados con datos des- y balanceados respectivamente. Cabe recalcar que a pesar de no contar con un rango de incertidumbre para estos valores, la mejora relativa podría considerarse significativa, pues la inclusión de tres transformaciones sencillas dentro del conjunto de transformaciones original, mejoró la detección de verdaderos positivos para todas las clases. De esta manera, podemos afirmar que, el conjunto de transformaciones juega un papel fundamental y además, la elección de ésta debe realizarse de acuerdo al problema en cuestión y las características de los datos disponibles.



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	87.86 %	89.94 %	88.89 %
1	59.04 %	63.40 %	61.15 %
2	68.60 %	75.41 %	71.84 %
3	81.94 %	52.78 %	64.21 %
4	30.24 %	91.46 %	45.45 %
macro avg	65.54 %	74.60 %	66.31 %
weighted avg	74.93 %	72.16 %	72.38 %
Exactitud	72.16 %		

(d) Reporte de clasificación

Figura 6-5: Métricas de evaluación obtenidas para el experimento de referencia modificado usando el codificador como inicialización y conjuntos de validación y prueba balanceados.

6.3. Experimentos transformaciones

Durante esta sección se dará, únicamente, enfoque a la métrica de exactitud, si bien es cierto que, las métricas ofrecidas con anterioridad son, también relevantes, la inclusión de las mismas dificultará el flujo y lectura de los datos relevantes para esta sección. De esta forma, solamente se presentarán el conjunto de métricas para los dos mejores experimentos y el experimento 4, el cual corresponde a las transformaciones originales.

La figura 6-6 muestra la distribución de clases para el conjunto de ajuste fino, utilizado durante todos los experimentos de esta sección. Mientras que las tablas del apéndice A-1, A-2, A-3 y A-4 muestran la pérdida y exactitud en los conjuntos de válidación entrenamiento y prueba para cada uno de los experimentos, cuyo ajuste fino se realizó con una tasa de aprendizaje de 5×10^{-3}, 2×10^{-3}, 1×10^{-3} y 7×10^{-4} respectivamente.

Puesto que los valores de la exactitud en los datos de prueba difieren entre sí debido a la tasa de aprendizaje empleada, se consideraron como los dos mejores experimentos, aquellos para los cuales el promedio de dicha exactitud sea mayor y su desviación estándar sea pequeña. Para este segundo criterio, se piensa que si la desviación estándar es pequeña, esto implica que la calidad de las representaciones, provenientes del preentrenamiento y por ende dependientes de las transformaciones aplicadas, puede considerarse más uniforme con respecto aquellos promedios con desviaciones mayores, pues el rango de tasa de aprendizaje en el cual se realizaron dichos experimentos es pequeño y por ende, los resultados no deberían variar significativamente. Con esto en mente, la tabla 6-1 muestra el promedio y desviación estándar de la exactitud en los datos de prueba de cada uno de los experimentos planteados.

Las figuras 6-7, 6-8 y 6-9 muestran las métricas de los experimentos E2, E12 y E4 respectivamente, conside-

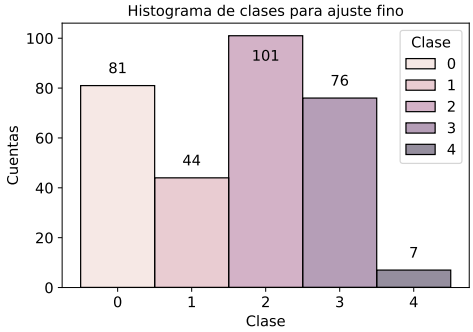


Figura 6-6: Distribución de clases en el conjunto Nair de ajuste fino para los experimentos.

Experimento	Promedio Exactitud[%]	Desviación estándar[%]
E1	65.65	0.32
E2	**68.38**	**0.52**
E3	67.71	1.08
E4	67.56	0.81
E5	68.25	0.84
E6	67.46	0.98
E7	65.81	0.53
E8	66.01	0.27
E9	67.15	0.62
E10	67.29	0.42
E11	66.80	0.38
E12	**68.33**	**0.34**

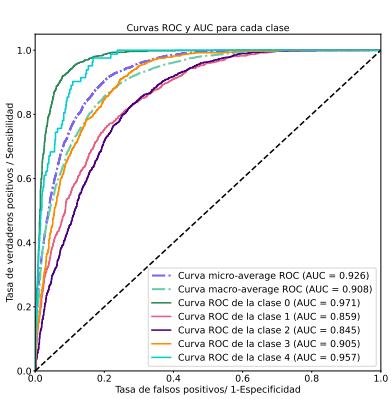
Tabla 6-1: Promedio y desviación estándar de los experimentos propuestos en la tabla 5-3 para todas las tasas de aprendizaje empleadas. En negritas se resaltan los dos mejores experimentos.

rando la tasa de aprendizaje para la cual la exactitud en prueba fue mayor.

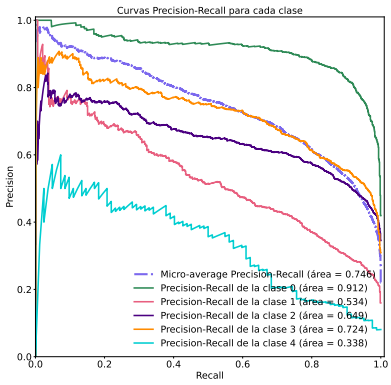
Los resultados obtenidos durante esta sección nos revelan que: la transformación extra más importante para todo el conjunto es el recorte central y el intervalo de área a cubrir en la trasformación de recorte aleatorio. Esto se justifica al comparar los resultados obtenidos por los experimentos E4, E5 y E8. En donde se observa que un recorte central, a proporción de 0.65 con un rango de área de [0.2, 1.0], incrementa el rendimiento promedio en $0.69\,\%$ y $2.24\,\%$ con respecto a su contraparte *original* (E4) y aquella con un recorte central más grande, pero con un rango de área menor (E8).

Los resultados obtenidos por el experimento 1, demuestran que, para nuestro caso, la aplicación de todas las transformaciones disponibles genera una especie de saturación, es decir, el aprendizaje de las representaciones durante la etapa autosupervisada se ve afectada debido a la complejidad de las transformaciones empleadas. Una posible explicación a este comportamiento, puede estar ligado fuertemente a los hiperparámetros contenidos dentro de cada una de las transformaciones.

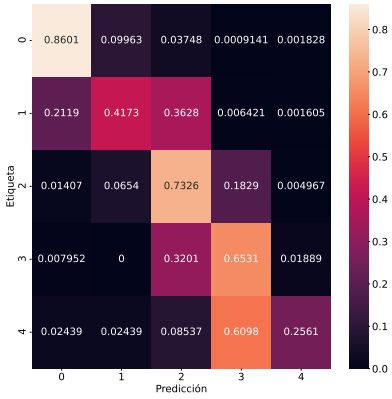
Al observar las curvas ROC-AUC y Precision-Recall de las figuras 6-9, 6-8 y 6-7 no se encuentran variaciones altamente significativas, salvo, para la clase 4. Lo mismo sucede con los valores $F1$, donde se encuentran diferencias significativas para las clases 4 y 1. Nótese que la diferencia entre el experimento E4 y los experimentos E2, E12, es de $4.08\,\%$ y $7.14\,\%$ respectivamente. Sin embargo al comparar los valores $F1$ correspondientes a la clase 4, así como las matrices de confusión, se tiene que las diferencias corresponden a $-20.97\,\%$ y $-16.77\,\%$, por lo que, es posible considerar que, las diferencias con respecto a la clase 1 se encuentran totalmente justificadas



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	85.54 %	86.01 %	85.78 %
1	57.78 %	41.73 %	48.46 %
2	59.76 %	73.26 %	65.82 %
3	70.42 %	65.31 %	67.77 %
4	42.86 %	25.61 %	32.06 %
macro avg	63.27 %	58.38 %	59.98 %
weighted avg	68.81 %	68.88 %	68.36 %
Exactitud	68.88 %		

(d) Reporte de clasificación

Figura 6-7: Métricas de evaluación obtenidas para el experimento E2 con una tasa de aprendizaje de 1×10^{-3} durante el ajuste fino.

si se considera cualquiera de los experimentos E2 o E12.

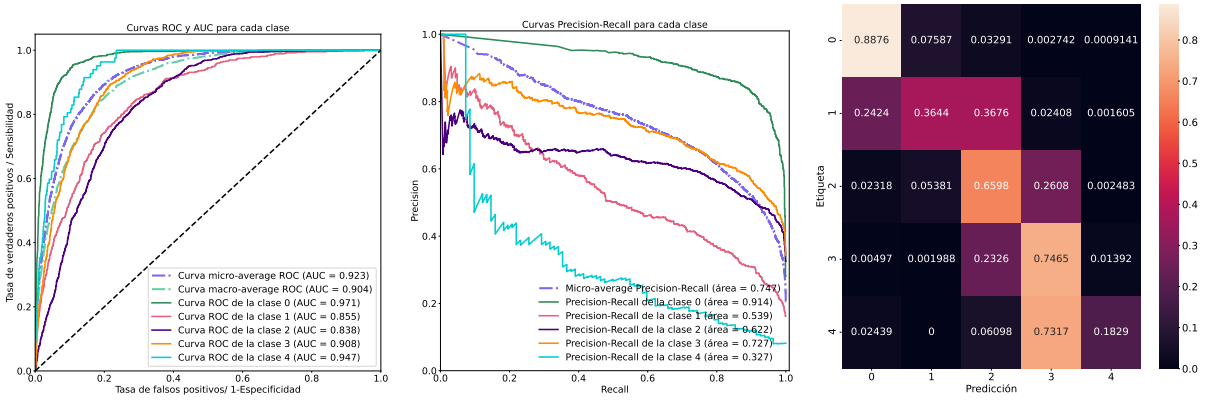
Notemos que los valores Recall, de las matrices de confusión para los experimentos E2, E4 y E12 son completamente diferentes entre sí, siendo el experimento E4, con el menor valor, por lo que es posible considerar que un conjunto de transformaciones un poco más complejo que el original refuerza este valor para la clase más desbalanceada. A pesar de que este hecho requiere ser demostrado formalmente, los experimentos presentados demuestran este hecho de forma empírica.

De esta serie de experimentos, implementados bajo las mismas condiciones, encontramos que el mejor conjunto de transformaciones aplicadas al método autosupervisado son las empleadas por el experimento E2, pues fue éste cuyas métricas son relativamente mejores a los otros. Es por esta razón por la cual los experimentos de referencia modificado y K-medias emplean dicho conjunto de transformaciones.

6.4. Experimentos K-medias

6.4.1. K-medias durante todo el entrenamiento

Las métricas de clasificación para el experimento que usó el algoritmo de agrupamiento durante todo el entrenamiento autosupervisado se muestran en la figura 6-10.



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	83.92 %	88.76 %	86.27 %
1	60.21 %	36.44 %	45.40 %
2	61.26 %	65.98 %	63.53 %
3	65.65 %	74.65 %	69.86 %
4	44.12 %	18.29 %	25.86 %
macro avg	63.03 %	56.82 %	58.18 %
weighted avg	68.02 %	68.80 %	67.73 %
Exactitud	68.80 %		

(d) Reporte de clasificación

Figura 6-8: Métricas de evaluación obtenidas para el experimento E12 con una tasa de aprendizaje de 2×10^{-3} durante el ajuste fino.

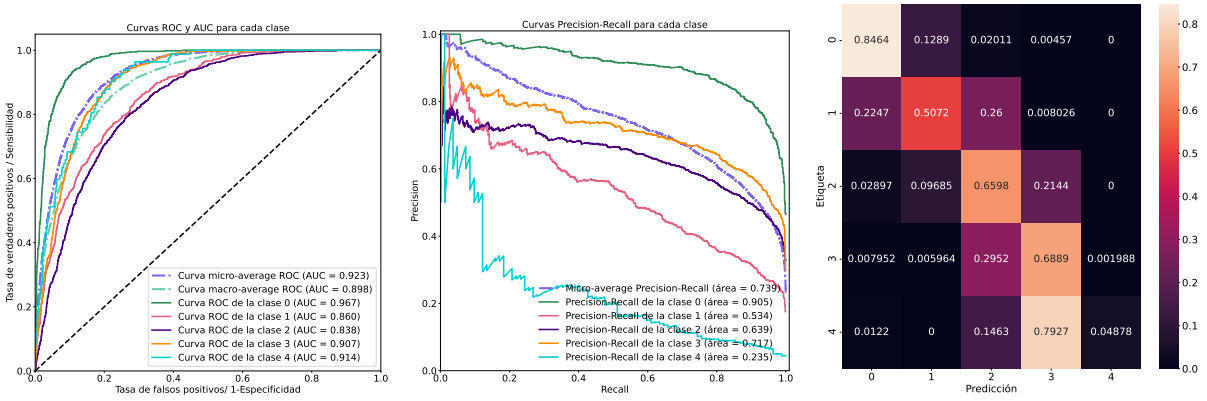
6.4.2. K-medias durante las primeras 50 épocas

Las métricas de clasificación para el experimento que usó el algoritmo de agrupamiento durante la primera mitad del entrenamiento autosupervisado se muestran en la figura 6-11.

6.4.3. K-medias durante las últimas 50 épocas

Las métricas de clasificación para el experimento que usó el algoritmo de agrupamiento durante la segunda mitad del entrenamiento autosupervisado se muestran en la figura 6-12.

La diferencia entre el área bajo las curvas ROC-AUC de los tres experimentos es insignificante, salvo para la representada por la clase 4, cuya diferencia es de al menos 0.022 unidades, siendo el experimento que considera el algoritmo de K-medias durante todo el entrenamiento autosupervisado con el mayor valor. En contraparte, se observan diferencias significativas en los valores del área bajo las curvas Precision-Recall para todas las clases, sin embargo, puesto que nuestro interés radica en las dos clases más desbalanceadas (4 y 1), nuevamente el experimento que emplea en su totalidad K-medias presenta un valor más alto para la clase 4 y altamente similar para la clase 1, comparados con respecto a los otros dos. Un argumento similar puede darse al comparar las matrices de confusión, en las cuales el primer método muestra una mejora significativa en cuanto a la clasificación correcta de la clase cuatro, con una diferencia de al menos 0.073 unidades y una mejora sobre la clase 1 de al menos 0.037 unidades. Más aún, los valores de $F1$ del reporte de clasificación para este mismo método, difieren en $9.23\,\%$. De esta manera, podemos inferir que el mejor método de estos tres se encuentra al emplear el algoritmo de K-medias durante todo el entrenamiento autosupervisado.



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	83.42 %	84.64 %	84.03 %
1	54.48 %	50.72 %	52.54 %
2	61.78 %	65.98 %	63.81 %
3	67.48 %	68.89 %	68.18 %
4	66.67 %	4.878 %	9.091 %
macro avg	66.77 %	55.02 %	55.53 %
weighted avg	68.08 %	68.18 %	67.55 %
Exactitud	68.18 %		

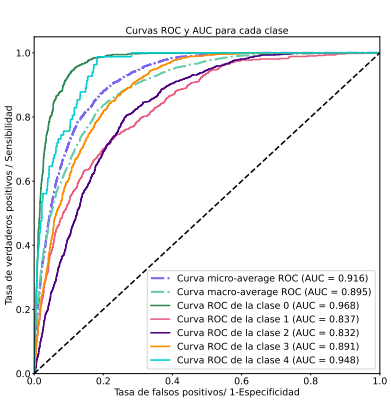
(d) Reporte de clasificación

Figura 6-9: Métricas de evaluación obtenidas para el experimento E4 con una tasa de aprendizaje de 7×10^{-4} durante el ajuste fino.

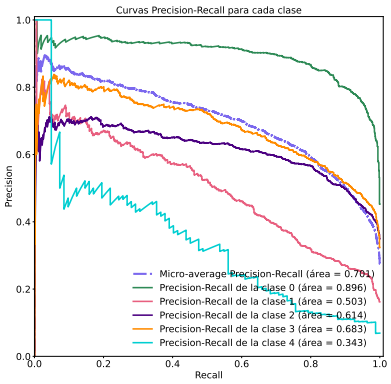
Una comparación similar entre este mejor experimento y el experimento 2 de la sección 6.3 revela que el hecho de emplear el algoritmo de K-medias, sí mejora las métricas de clasificación para las dos clases más desbalanceadas y se obtienen métricas con diferencias poco significativas para la tercer clase más desbalanceada (clase 3). A pesar de esta mejoría, también se observa una disminución significativa de las métricas al considerar las dos clases mayoritarias, lo cual podría estar correlacionado con la naturaleza continua entre clases de galaxias.

De especial mención y hecho que no debe olvidarse es la reselección o reinicio de muestras de entrenamiento, las cuales estaban limitadas por un valor umbral correspondiente a aproximadamente 15 mil imágenes por época, esto es, cada clase era balanceada, a través de las pseudoetiquetas, con la posibilidad de contener, a lo más, 3 mil imágenes. Este procedimiento, claramente, puede sesgar los resultados, sin embargo, al considerar 3 mil imágenes por cluster cuya cantidad es mayor a la presente en la clase mayoritaria de la figura 4-2 y suponer que la distribución de clusters es de tal forma que la distancia entre éstos es la adecuada, se esperaría, estadísticamente, que una proporción significativa de cada clase perteneciente al conjunto original, se encuentre dentro de cada reinicio de muestras.

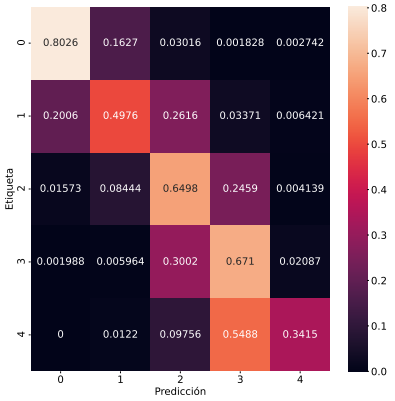
Un hallazgo interesante, que puede estar fuertemente relacionado con lo anteriormente mencionado, es que los valores de exactitud contrastiva se saturaban a un valor especifico de aproximadamente 68 %, siempre y cuando se utilice el algoritmo de K-medias, sugiriendo que, la reinicialización de muestras es efectivamente cambiante en el tiempo y además éste incrementa la dificultad de la tarea. La figura 6-13 ilustra este hecho para los tres experimentos mencionados.



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	85.74 %	80.26 %	82.91 %
1	51.93 %	49.76 %	50.82 %
2	60.81 %	64.98 %	62.82 %
3	64.90 %	67.10 %	65.98 %
4	45.90 %	34.14 %	39.16 %
macro avg	61.86 %	59.25 %	60.34 %
weighted avg	66.95 %	66.68 %	66.74 %
Exactitud	66.68 %		

(d) Reporte de clasificación

Figura 6-10: Métricas de evaluación obtenidas para el experimento que emplea K-medias durante todo el entrenamiento autosupervisado. La tasa de aprendizaje durante el ajuste fino fue de 5×10^{-3}.

6.5. Experimentos supervisados y efectividad del método autosupervisado

6.5.1. Método completamente supervisado desde cero

Las figuras 6-14 y 6-15 muestran las métricas de clasificación para los experimentos completamente supervisados desde cero cuyo entrenamiento constó de los conjuntos de validación y entrenamiento desbalanceados y balanceados respectivamente.

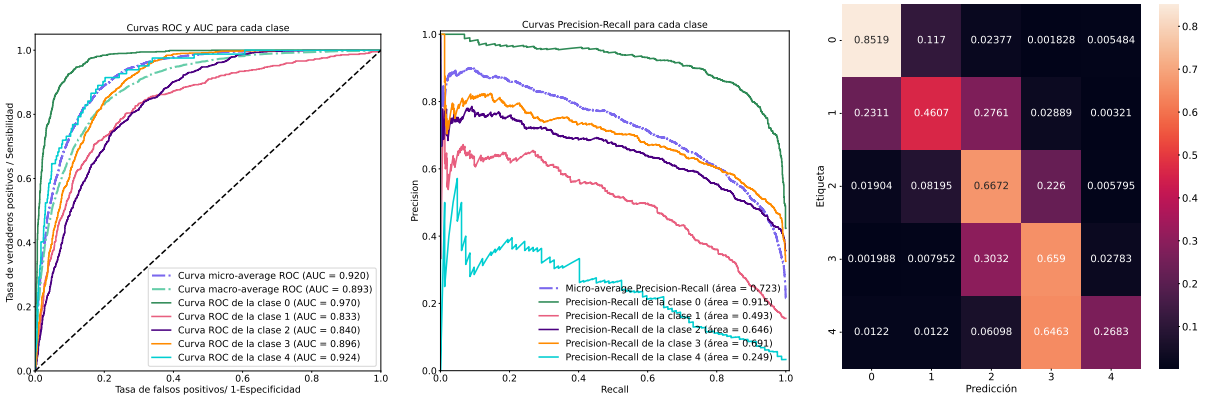
6.5.2. Método completamente supervisado con pesos de ImageNet

Las figuras 6-16 y 6-17 muestran las métricas de clasificación para los experimentos completamente supervisados con pesos de ImageNet cuyo entrenamiento constó de los conjuntos de validación y entrenamiento desbalanceados y balanceados respectivamente.

6.5.3. Efectividad del método autosupervisado

Puesto que este experimento fue diseñado para comprobar que el aprendizaje de las representaciones, durante la etapa autosupervisada, aporta información relevante de las imágenes durante el ajuste fino, únicamente se presentará el reporte de clasificación (ver tabla 6-2).

De los métodos completamente supervisados puede observarse que el mejor modelo es el que emplea los pesos aprendidos para la tarea de clasificación en el conjunto de ImageNet. Sin embargo, tanto para los casos



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	84.57 %	85.19 %	84.88 %
1	54.88 %	46.07 %	50.09 %
2	61.34 %	66.72 %	63.92 %
3	65.71 %	65.90 %	65.81 %
4	33.85 %	26.83 %	29.93 %
macro avg	60.07 %	58.14 %	58.92 %
weighted avg	67.20 %	67.53 %	67.26 %
Exactitud	67.53 %		

(d) Reporte de clasificación

Figura 6-11: Métricas de evaluación obtenidas para el experimento que emplea K-medias durante la primera mitad del entrenamiento autosupervisado. La tasa de aprendizaje durante el ajuste fino fue de 5×10^{-3}.

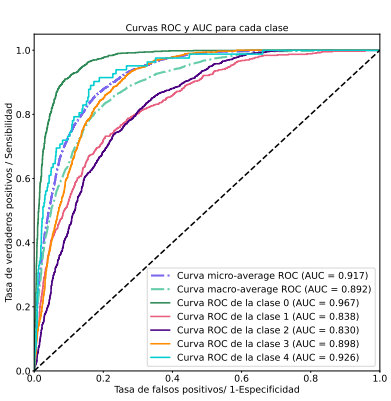
Clase	Precision	Recall	F1
0	0.00 %	0.00 %	0.00 %
1	0.00 %	0.00 %	0.00 %
2	30.19 %	97.93 %	46.15 %
3	56.38 %	5.27 %	9.64 %
4	0.00 %	0.00 %	0.00 %
macro avg	17.31 %	20.64 %	11.16 %
weighted avg	23.22 %	30.80 %	16.31 %
Exactitud	30.80 %		

Tabla 6-2: Reporte de clasificación para el codificador con pesos inicializados aleatoriamente y congelados.

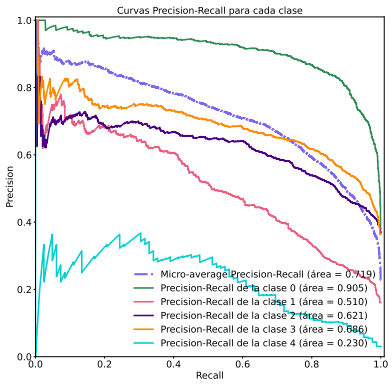
balanceados como para los desbalanceados, la diferencia entre las métricas no es significativamente grande y éstas pueden ser explicadas al considerar que el modelo en ImageNet fue preentrenado con al rededor de 1.3 millones de imágenes, que comparado con el conjunto de Nair, es aproximadamente 130 veces mayor, lo cual dota a la red de un poder expresivo mayor con respecto a la entrenada únicamente con el conjunto de Nair.

Por otro lado, note que el uso de conjuntos de datos balanceados mejora, significativamente, las predicciones correctas sobre las clases 4 y 1, a pesar de ello, se observa una disminución en las predicciones correctas sobre las clases 0, 2 y 3. Este hecho puede ser asociado a la continuidad de las clases, pues observemos que en el caso balanceado, donde la distribución es uniforme, el porcentaje de error con respecto a clases contiguas para cualquiera de ellas, es de, al menos, 8 %.

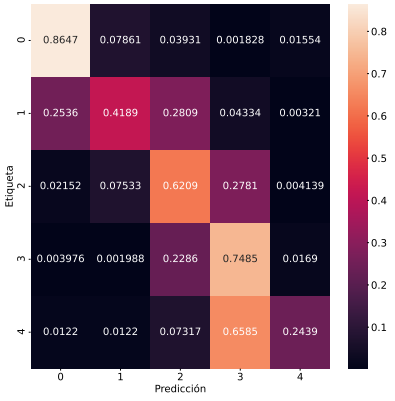
Considérese ahora, los resultados presentados en las tablas 1-1 y 3-2, en particular los obtenidos por [3,46,47]



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	83.35 %	86.47 %	84.88 %
1	59.18 %	41.89 %	49.06 %
2	62.29 %	62.09 %	62.19 %
3	64.25 %	74.85 %	69.15 %
4	32.79 %	24.39 %	27.97 %
macro avg	60.37 %	57.94 %	58.65 %
weighted avg	67.44 %	68.03 %	67.38 %
Exactitud	68.03 %		

(d) Reporte de clasificación

Figura 6-12: Métricas de evaluación obtenidas para el experimento que emplea K-medias durante la segunda mitad del entrenamiento autosupervisado. La tasa de aprendizaje durante el ajuste fino fue de 5×10^{-3}.

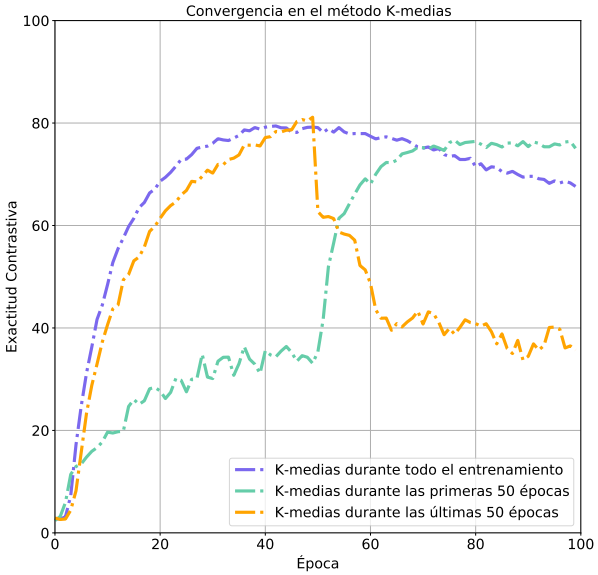
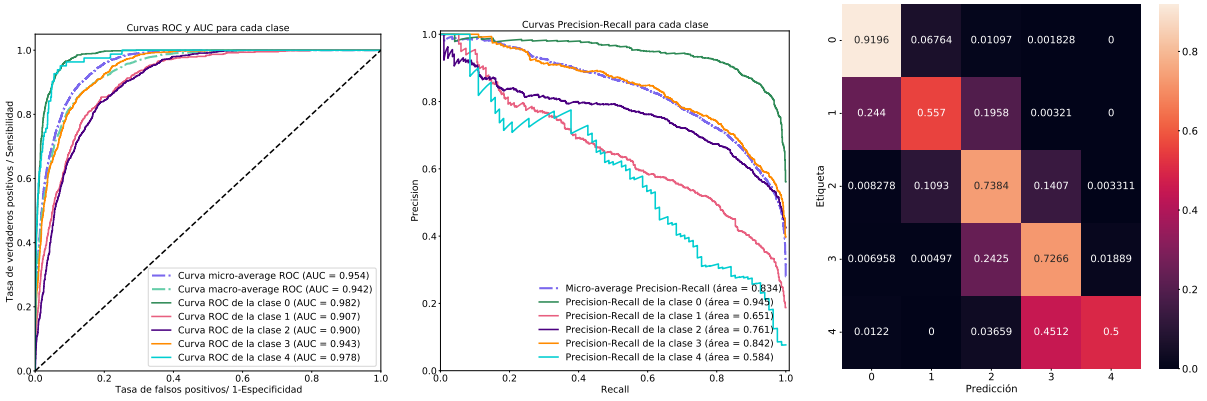


Figura 6-13: Distribución de clases en el conjunto Nair de ajuste fino para los experimentos.

para 3, 5 y 7 clases, los valores obtenidos durante esta fase supervisada, se encuentran dentro del rango esperado, establecido por estudios anteriores, a excepción de [47]. Sin embargo, no es posible ofrecer una comparación



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	85.54 %	91.96 %	88.63 %
1	62.19 %	55.70 %	58.76 %
2	70.07 %	73.84 %	71.91 %
3	77.60 %	72.66 %	75.05 %
4	64.06 %	50.00 %	56.16 %
macro avg	71.89 %	68.83 %	70.10 %
weighted avg	74.83 %	75.18 %	74.89 %
Exactitud	75.18 %		

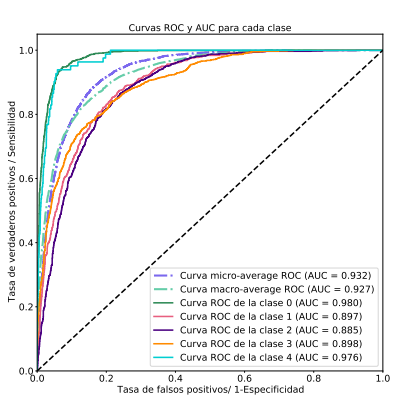
(d) Reporte de clasificación

Figura 6-14: Métricas de evaluación obtenidas para el experimento completamente supervisado usando conjuntos de validación y entrenamiento desbalanceados.

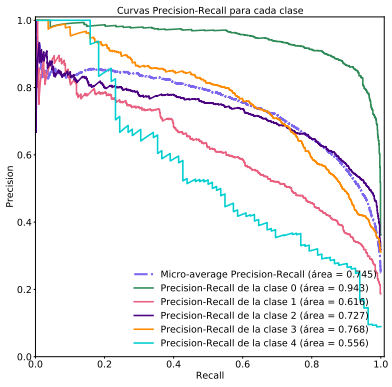
directa entre los resultados presentados y los estudios [3, 47], pues en éstos se emplea un esquema de clasificación que dista del empleado en este trabajo. Si bien es cierto que en [47] las métricas presentadas son cercanas al 100 %, estos valores excelentes pueden deberse a dos motivos principales: El cambio o modificación de la arquitectura ResNet, que para este estudio quedó fuera de nuestro alcance, y el uso de muestras *limpias* cuyas etiquetas se encuentran regidas por el esquema de clasificación de Galaxy Zoo 2, la cual, podría estar mejor definida con respecto a la continuidad y cuantización de clases. En contraparte, la comparación directa con [46] nos muestra que los resultados obtenidos por ambas partes son altamente similares entre sí, cuyas diferencias se deben al preprocesamiento de datos, así como la diferencia entre el uso de los datos desbalanceados.

A pesar de ser un experimento sumamente sencillo, la efectividad del método autosupervisado queda demostrada al observar el bajo rendimiento ofrecido por el modelo iniciado aleatoriamente con respecto al codificador como extractor de características empleado durante el experimento de referencia.

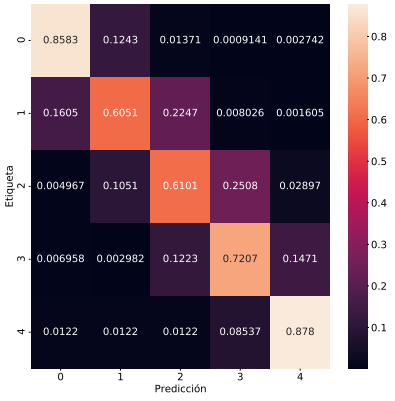
Aún cuando la intención de este trabajo no radica en comparar los métodos supervisados con los senisupervisados, nótese que las métricas de evaluación para el método supervisado desde cero (figuras 6-14 y 6-15), así como el método con pesos de ImageNet (figuras 6-16 y 6-17) son altamente similares entre las producidas por los experimentos de referencia y modificado (figuras 6-2, 6-3, 6-4 y 6-5). Aunque este hecho ya ha sido demostrado por [12, 27], a través de ambas implementaciones ha sido posible confirmarlo.



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	89.17 %	85.83 %	87.47 %
1	58.54 %	60.51 %	59.51 %
2	72.54 %	61.01 %	66.28 %
3	69.64 %	72.07 %	70.84 %
4	27.80 %	87.80 %	42.23 %
macro avg	63.54 %	73.45 %	65.26 %
weighted avg	73.26 %	71.02 %	71.66 %
Exactitud	71.02 %		

(d) Reporte de clasificación

Figura 6-15: Métricas de evaluación obtenidas para el experimento completamente supervisado usando conjuntos de validación y entrenamiento balanceados.

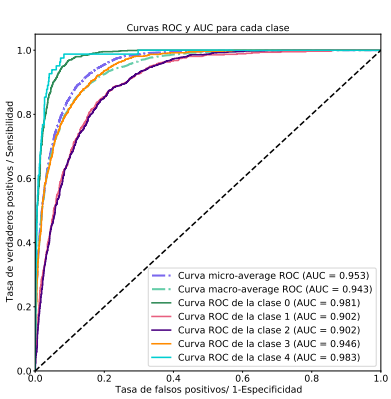
6.6. Experimentos con otros codificadores (EfficientNetV2 & DenseNet-161)

Dado que el cambio de codificador ResNet-50 por codificadores más recientes, que han superado el estado del arte, sigue únicamente fines cuantitativos, que ayudarán a comprender la naturaleza y dificultad del problema al cual se enfrenta este trabajo, únicamente nos centraremos en discutir las métricas de evaluación para ambos codificadores al compararlas directamente con los resultados del experimento 2 (E2) presentado en la sección 6.3 ya que comparten el mismo tratamiento de datos e hiperparámetros. Así, La tabla 6-3 muestra los valores de exactitud y pérdida para el método autosupervisado de ambos codificadores, mientras que las figuras 6-18 y 6-19 corresponden a las métricas obtenidas por las arquitecturas EfficientNetV2 de tamaño medio y DenseNet-161 durante el ajuste fino respectivamente.

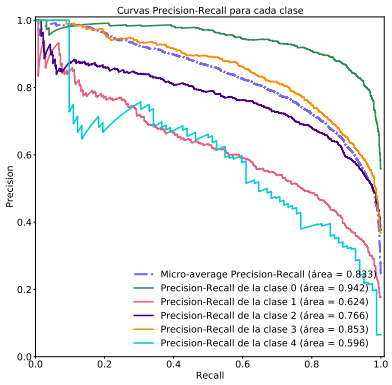
Codificador	Exactitud Contrastiva [%]	Pérdida Contrastiva [u.a.]
EfficientNetV2 m	92.74 %	1.10
DenseNet-161	98.02 %	0.113

Tabla 6-3: Métricas al final del entrenamiento autosupervisado (100 épocas) para los codificadores EfficientNetV2 de tamaño medio y DenseNet-161.

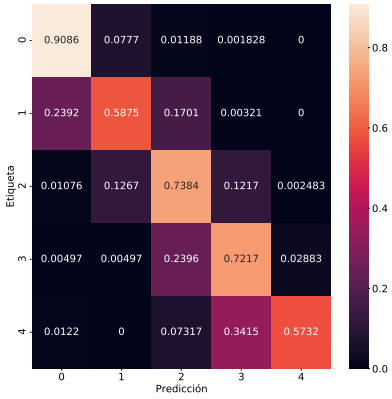
Al comparar la matriz de confusión entre las figuras 6-18 y 6-7, se observa que para el caso del codificador efficientNetV2 las predicciones correctas para las clases 0 y 3 mejoran con respecto a su contraparte ResNet-50, sin embargo esta mejora no es significativamente mayor como para considerar a esta arquitectura como un



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	85.54 %	90.86 %	88.12 %
1	60.10 %	58.75 %	59.42 %
2	70.91 %	73.84 %	72.34 %
3	80.22 %	72.17 %	75.98 %
4	59.49 %	57.32 %	58.39 %
macro avg	71.25 %	70.59 %	70.85 %
weighted avg	75.32 %	75.38 %	75.26 %
Exactitud	75.38 %		

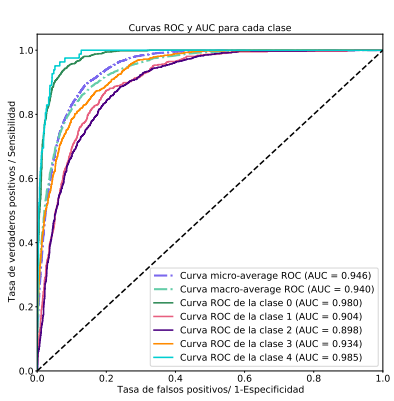
(d) Reporte de clasificación

Figura 6-16: Métricas de evaluación obtenidas para el experimento completamente supervisado usando conjuntos de validación y entrenamiento desbalanceados.

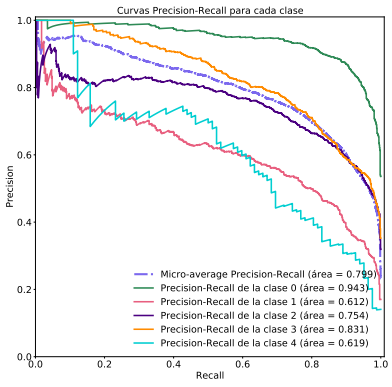
avance, pues, las clases 1, 2 y 4 se ven severamente deterioradas, obteniendo una discrepancia de hasta 25 %. El mismo panorama se mantiene al observar los valores micro- y macro-averange de las curvas ROC-AUC y Precision-Recall siendo más altos para el codificador ResNet-50, en particular, obsérvese el área bajo ambas curvas correspondiente a la clase 4 donde se logra una discrepancia significativa a favor del codificador original. Ahora bien, al sopesar los valores $F1$ entre ambos codificadores, el experimento 2 obtiene valores mayores, excepto para la clase 3 cuya diferencia es de, apenas, 2.26 %. Bajo estas condiciones, es posible descartar el codificador EfficientNetV2 de tamaño medio como un posible sustituto del codificador original.

Comparemos ahora las métricas entre el codificador DenseNet-161 y ResNet-50; la discrepancia entre las predicciones correctas, en las matrices de confusión, no son significativamente relevantes, excepto para la clase 4, cuya diferencia es de 20.73 % a favor del codificador original. Con respecto a los valores de las curvas ROC-AUC y Precision-Recall para ambos casos se observa que éstos son altamente similares entre sí, cuya diferencia más significativa es de 0.076 unidades. La misma tendencia se mantiene al comparar los valores $F1$, excepto para la clase 4. A pesar de que las métricas benefician al codificador Resnet-50, únicamente considerando la clase 4, en algunas otras clases el codificador DenseNet-161 muestra ser mejor (por ejemplo la clase 3). De esta manera, se puede conjeturar que, ambas arquitecturas son equiparables entre sí en cuanto al rendimiento sobre la tarea de clasificación morfológica de galaxias.

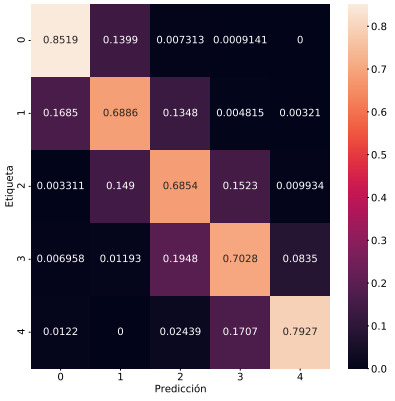
Un hecho interesante, que se observó durante el entrenamiento autosupervisado empleando el codificador DenseNet-161 fue el tiempo de convergencia, considerando como unidad de medida una época, a éste codificador le tomó 14 épocas alcanzar un valor de 81.60 en la exactitud contrastiva, que para el caso del codificador ResNet-50, bajo las mismas condiciones, requiere de 49 épocas para alcanzar un valor similar (ver figura 6-20). A pesar de que las tablas A-1, A-2, A-3 y A-4 demuestran que valores altos en la exactitud contrastiva no aseguran un



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	88.85 %	85.19 %	86.98 %
1	55.43 %	68.86 %	61.42 %
2	74.06 %	68.54 %	71.20 %
3	77.78 %	70.28 %	73.84 %
4	39.88 %	79.27 %	53.06 %
macro avg	67.20 %	74.43 %	69.30 %
weighted avg	75.43 %	73.79 %	74.27 %
Exactitud	73.79 %		

(d) Reporte de clasificación

Figura 6-17: Métricas de evaluación obtenidas para el experimento completamente supervisado usando conjuntos de validación y entrenamiento balanceados.

rendimiento mejor, podría estudiarse el impacto sobre el rendimiento en función de la cantidad de épocas en el entrenamiento autosupervisado. En caso de comprobarse la hipótesis de la convergencia acelerada gracias al codificador DenseNet-161, los experimentos, podrán realizarse de forma más rápida y con costes reducidos.

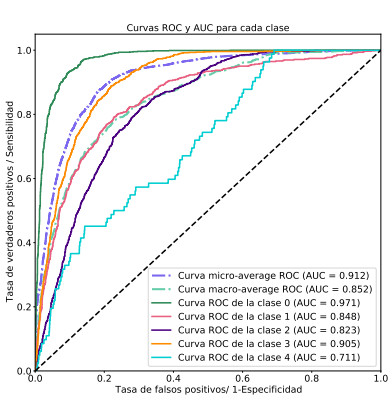
6.7. Visualización

Puesto a que esta sección es de índole cualitativa, cuyo fin consiste en dar un panorama general y visual del comportamiento de las redes presentadas con anterioridad, se presentarán únicamente aquellas imágenes consideradas más relevantes, mientras que el conjunto completo de éstas podrá encontrarse en el apéndice A. El orden de aparición de cada etiqueta para cada imagen particular, se debe a un ordenamiento descendiente. Por otro lado, las imágenes de cada etiqueta se muestran con un valor de transparencia definido y en escala de grises. La sensación de color se debe a los valores SHAP cercanos a cero.

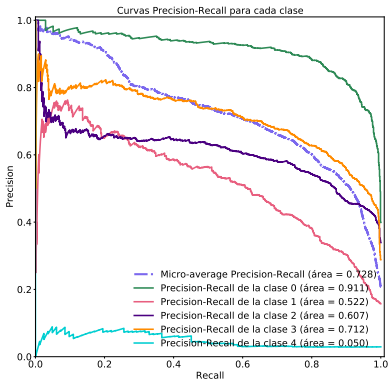
6.7.1. SHapley Additive exPlanations

La visualización de los valores SHAP para un conjunto de imágenes se muestran en las figuras 6-21, 6-22, 6-23, 6-24 y 6-25, resultantes del modelo semisupervisado desbalanceado con el conjunto de transformaciones extra (experimento de referencia modificado). El subconjunto de imágenes mostrado, así como el presentado en el apéndice A, fueron elegidos aleatoriamente del conjunto Nair, de tal forma que todas las clases contengan el mismo número de muestras.

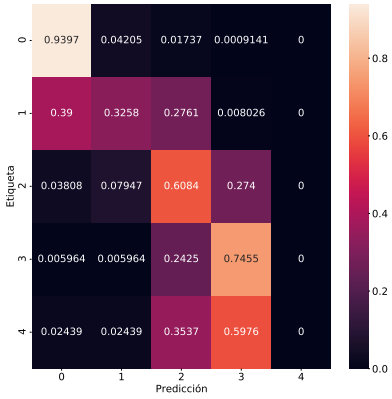
De las imágenes SHAP, es posible inferir las regiones de interés aprendidas por el modelo para cada una de



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	77.58 %	93.97 %	84.99 %
1	57.51 %	32.58 %	41.60 %
2	61.30 %	60.84 %	61.07 %
3	66.02 %	74.55 %	70.03 %
4	0.00 %	0.00 %	0.00 %
macro avg	52.48 %	52.39 %	51.54 %
weighted avg	65.08 %	67.68 %	65.57 %
Exactitud	67.68 %		

(d) Reporte de clasificación

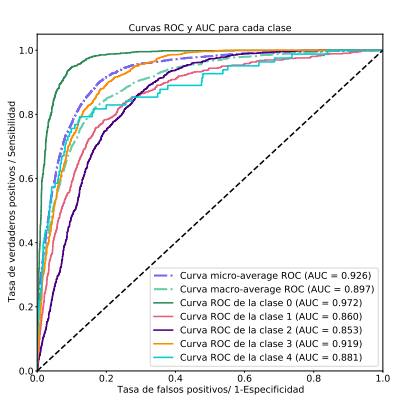
Figura 6-18: Métricas de evaluación obtenidas para el codificador EfficientNetV2 de tamaño medio usando conjuntos de validación y entrenamiento desbalanceados.

las clases. Obsérvese que para las clases 0 y 1, (6-21, 6-22) dichas regiones se encuentran, en su mayoría, concentradas en el centro de cada galaxia y/o sus zonas con mayor intensidad luminosa. Mientras que, las clases 2 y 3, (6-23, 6-24) las regiones más relevantes para el modelo abarcan desde su centro hasta la posible presencia de brazos, de esta forma, se considera la galaxia en su totalidad. Notemos que los valores SHAP para la imagen más ”ruidosa” de nuestro conjunto (última imagen de 6-24) son dispersos, sugiriendo que el modelo no logra encontrar un patrón bien definido, razón por la cual, su clasificación, errónea, contiene una probabilidad baja. Al considerar la vista y la apariencia morfológica de esta imagen, el lector la reconocerá más similar a las morfologías presentadas por la clase 2. Un argumento similar puede brindarse al considerar la baja probabilidad de la segunda imagen en 6-24.

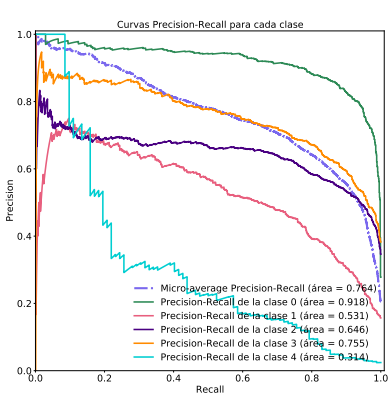
Por otro lado, de las imágenes de la clase 4, se observa, también, un patrón particular, para este caso, las regiones de decisión provistas por el modelo, se concentran en el centro, sin embargo, los alrededores del objeto de interés son tomados en cuenta. Este hecho puede deberse a la naturaleza ”irregular” de la clase, sin embargo, también debe considerarse que esta fue la clase más desbalanceada, por lo que es posible se trate de algún artefacto generado por la red.

Nótese que para algunos casos la probabilidad de pertenencia de clase para el valor de predicción es relativamente bajo o comparable a su vecino más cercano. Para estas muestras, se observa que la morfología, así como la apariencia de la imagen, es altamente similar entre ambas clases.

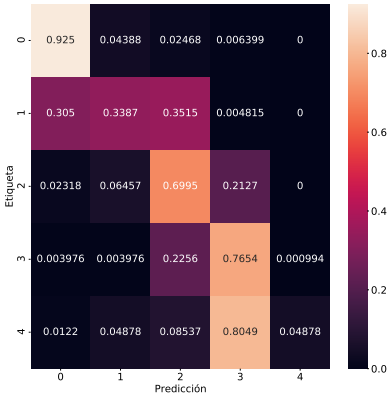
De este pequeño subconjunto es posible destacar, que las predicciones realizadas por el modelo, hayan sido correctas o no, son muy factibles, aún cuando las imágenes a predecir sean ”ruidosas”.



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	81.94 %	92.50 %	86.90 %
1	61.16 %	33.87 %	43.60 %
2	63.77 %	69.95 %	66.72 %
3	69.81 %	76.54 %	73.02 %
4	80.00 %	4.88 %	9.20 %
macro avg	71.34 %	55.55 %	55.89 %
weighted avg	70.17 %	70.82 %	69.04 %
Exactitud	70.82 %		

(d) Reporte de clasificación

Figura 6-19: Métricas de evaluación obtenidas para el codificador DenseNet-161 usando conjuntos de validación y entrenamiento desbalanceados.

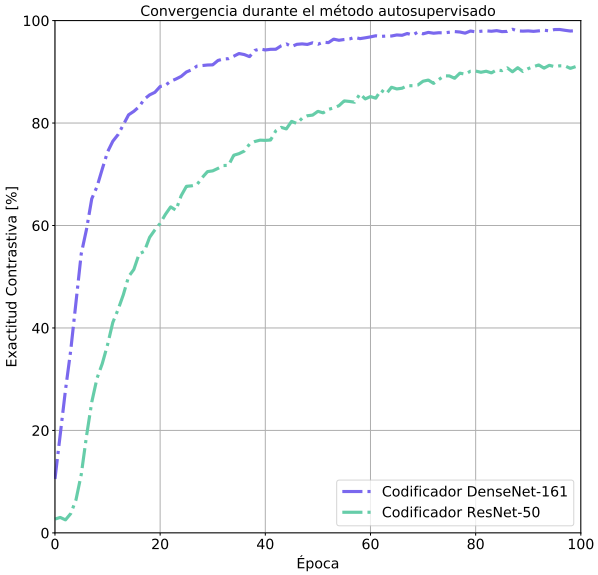


Figura 6-20: Tiempo de convergencia en el entrenamiento autosupervisado entre los codificadores ResNet-50 y DenseNet-161 bajo las miasmas condiciones.

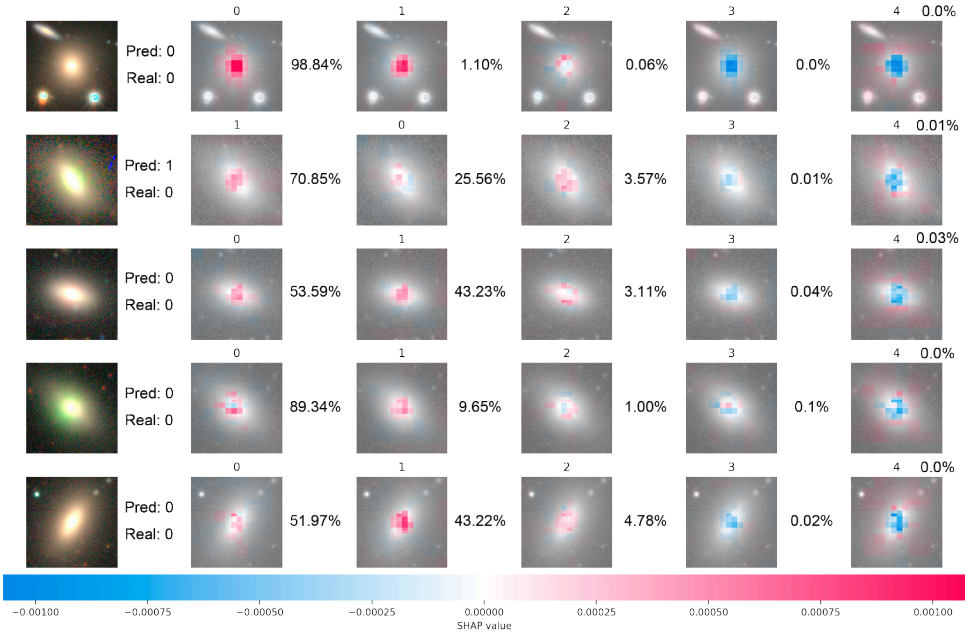


Figura 6-21: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 0 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

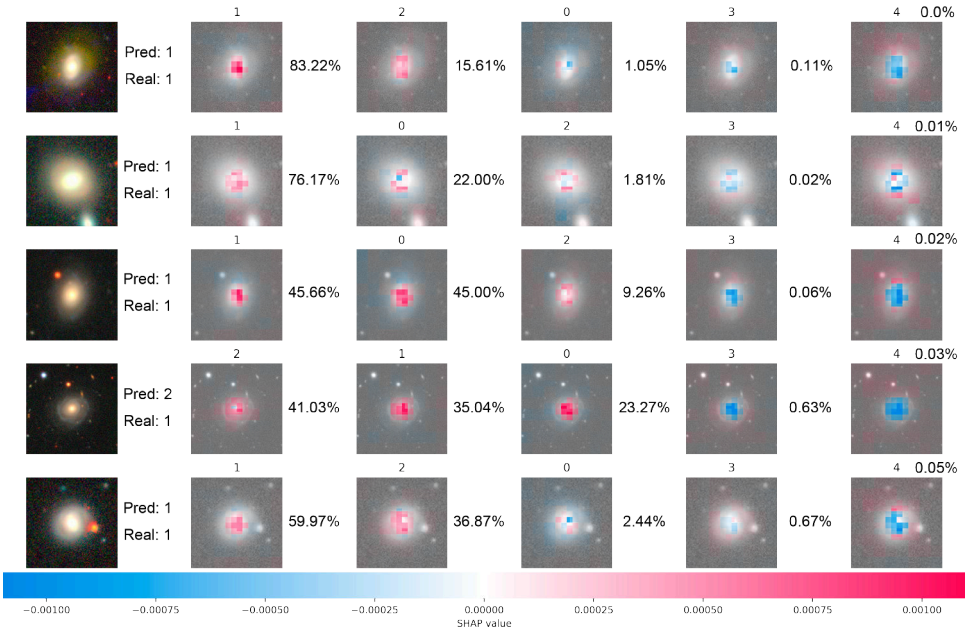


Figura 6-22: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 1 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

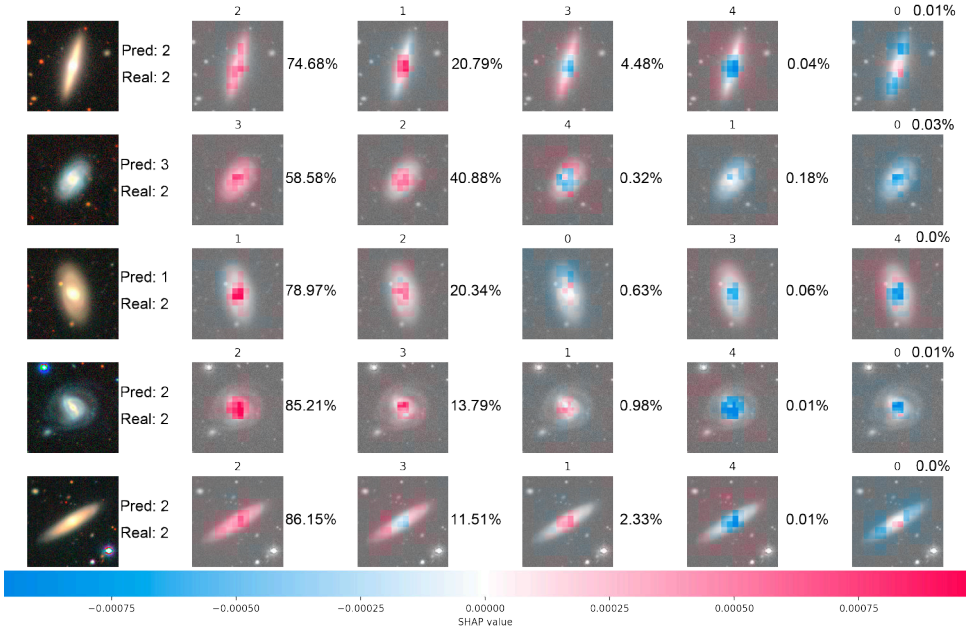


Figura 6-23: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 2 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

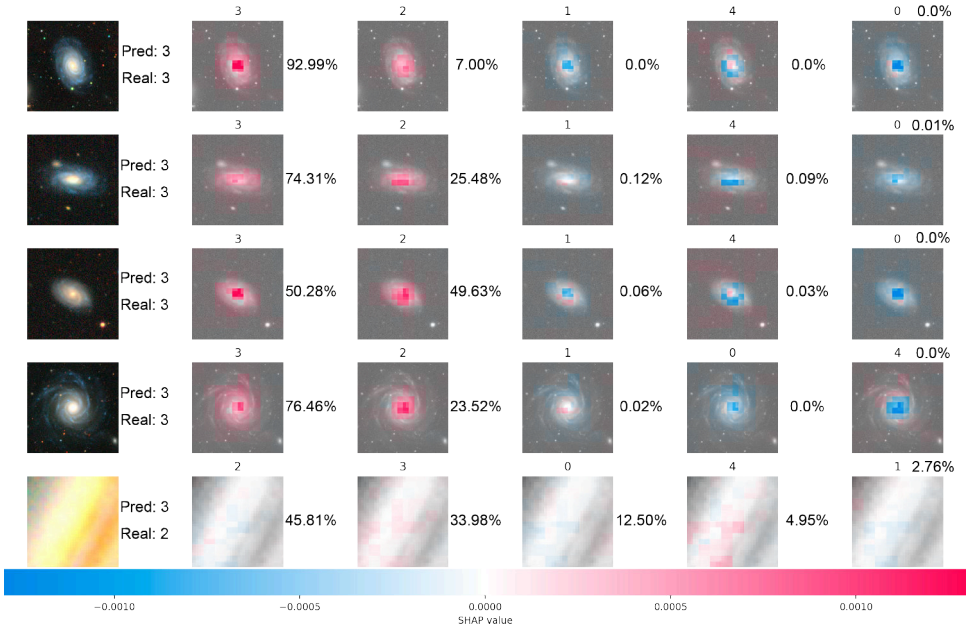


Figura 6-24: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 3 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

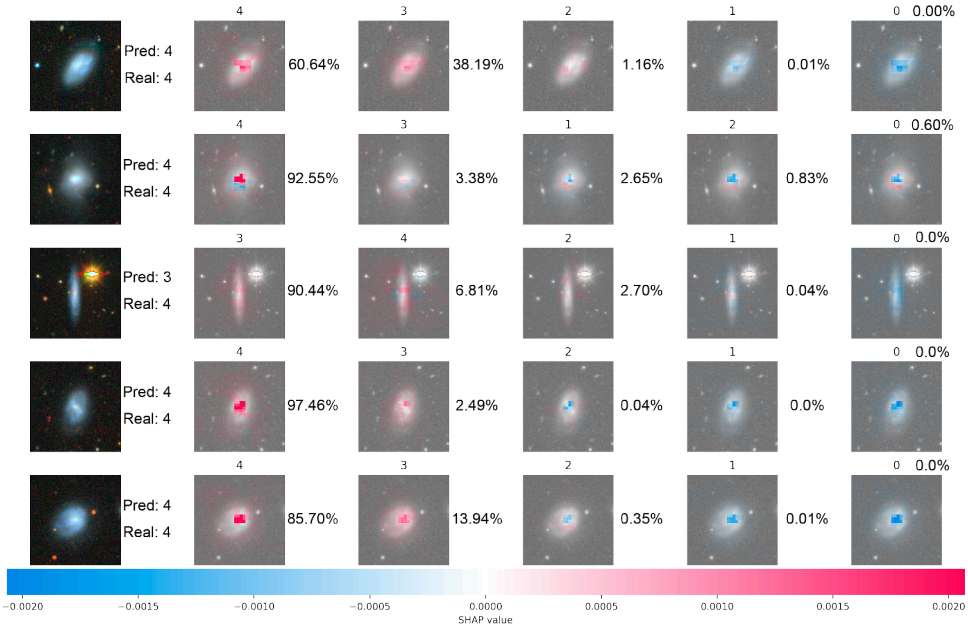


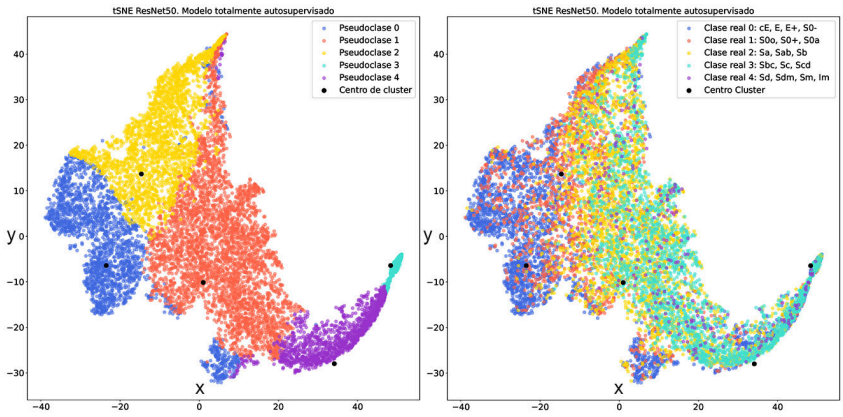
Figura 6-25: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 4 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

6.7.2. Distribución de clases por K-medias

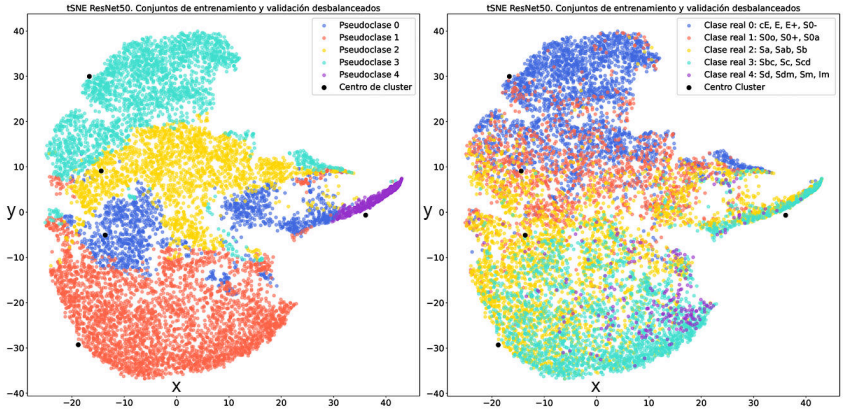
La distribución de clases a través de las etiquetas reales provistas en el conjunto Nair y las generadas por el algoritmo de K-medias se muestran en la figura 6-26. 6-26a muestra las distribuciones de clases provista por el codificador de la red entrenada por el método autosupervisado, mientras que 6-26b representa las distribuciones provistas por el codificador de la red semisupervisada. Los resultados corresponden al método Experimento de referencia modificado (6.2) descrito anteriormente. Cabe mencionar, que estos resultados corresponden al uso de datos desbalanceados. El uso de los datos balanceados no presentan variaciones importantes, por lo que no serán presentados.

Como es de esperar, la distribución de clases generadas por K-medias es completamente uniforme. A pesar de ello, los grupos se encuentran separados por una distancia muy pequeña, lo cual podría ser explicado por la naturaleza continua de las clases. Más aún, al observar las distribuciones reales, se destaca que existe una superposición, situación que también es esperada, pues la transición continua genera este efecto. Por otro lado, dicha superposición no debe ser únicamente atribuida a dicha transición, si no, también, a la red y sus métodos de entrenamiento, ya que, recordemos, el método autosupervisado separa espacialmente aquellas muestras negativas, mientras que agrupa aquellas muestras positivas, de esta manera, la distribución de clases para un clasificador perfecto, resultaría en un conjunto de grupos bien definidos, cuyas distancias entre sí son mayores que cero. Debe considerarse, también, que las técnicas de reducción de dimensionalidad, como lo es TSNE, suelen perder información con tal de realizar una visualización en 2 o 3 dimensiones.

Ahora bien, al comparar directamente las distribuciones de clase reales, se observa que el método semisupervisado logra separar, relativamente, mejor cada una de las clases. En particular, las clases 4 y 2 (morado, amarillo) se yuxtaponen con la clase 3 (turquesa), mientras que las clases 1 y 2 (naranja y amarillo) se encuentran en las regiones de la clase 0 (azul).



(a) Distribución de clases a través del codificador de la red completamente autosupervisada. Izquierda: distribución de las pseudoclases generadas por el algoritmo de K-medias. Derecha: Distribución real de las clases.



(b) Distribución de clases a través del codificador de la red semisupervisada. Izquierda: distribución de las pseudoclases generadas por el algoritmo de K-medias. Derecha: Distribución real de las clases.

Figura 6-26: Comparación de la distribución de las clases a través del algoritmo de K-medias con respecto a la distribución real.

Para el caso semisupervisado, se observa que la distribución de grupos generada por K-medias es similar en forma y tamaño a la distribución real (los colores no coinciden debido a la asignación de pseudoclases). Esto no sucede así con el método autosupervisado, excepto, quizás, para la clase 0.

Como se ha visto, la comparación de las distribuciones de clase tanto reales como las generadas por K-medias, son una herramienta poderosa para la evaluación e inferencia de las clases reales, pues, al menos, para el método semisupervisado, este análisis ayuda a obtener una idea intuitiva, aunque no certera, de dicha distribución. Por ende, es posible usar este mismo método para estimar la distribución de clases en el conjunto DESI no etiquetado. De esta manera, las figuras 6-27 y 6-28 muestran la distribución de pseudoclases generadas por el algoritmo K-medias para 5, 14 y 26 grupos a través de las redes semi- y auto- supervisadas respectivamente.

Para todos los casos se observa que la estimación de la distribución de clases es no balanceada, algo que podía intuirse desde un principio. Nótese que el hecho de incrementar las clases, no produce una mejor separación de los grupos e incluso aparenta añadir complejidad para la detección de clases.

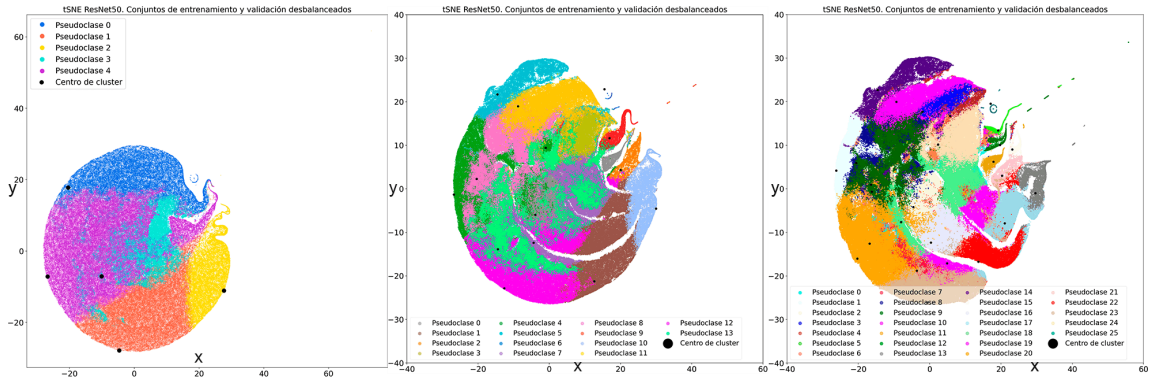


Figura 6-27: Distribución de pseudoclases producidas por K-medias usando el modelo semisupervisado y el conjunto de imágenes DESI, para izquierda: 5, medio: 14 y derecha: 26 grupos o clases.

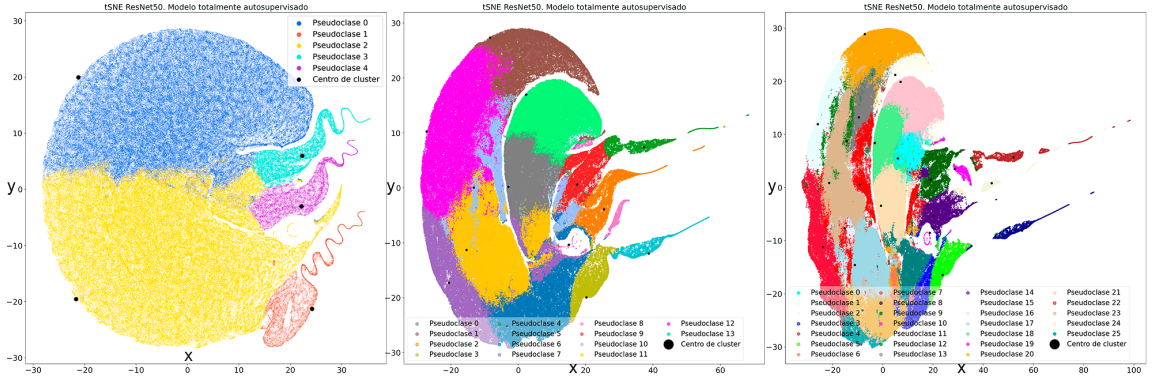


Figura 6-28: Distribución de pseudoclases producidas por K-medias usando el modelo autosupervisado y el conjunto de imágenes DESI, para izquierda: 5, medio: 14 y derecha: 26 grupos o clases.

Para responder a la pregunta ¿Cómo es posible saber si la división a cinco clases es la correcta y/o óptima? Se ajustaron los métodos heurísticos de clusterización *elbow*, así como el método de Silhouette con el fin de conocer la cantidad de clases correcta. Las gráficas obtenidas por estos métodos se muestran en la figura 6-29, en donde puede apreciarse que para el método de dispersión de grupos, la cantidad de clases óptima es 5. Por otro lado, el método Silhouette, encuentra como óptimos dos clases, sin embargo, este resultado es descartado al tratarse de un conjunto muy pequeño y ampliamente estudiado con resultados excelentes, de esta forma, la cantidad óptima es de 6 grupos. De esta forma, podemos, asegurar que la división a 5 clases es subóptima, ya que si bien, estos métodos no establecieron un único valor, fue posible establecer que el valor óptimo se encuentra entre 5 y 6.

6.7.3. Cuadrículas de activación

Para las cuadrículas de activación se analizaron 5 imágenes seleccionadas de manera pseudoaletaoria, procurando seleccionar aquellas que se consideraron más representativas de cada clase. Puesto a que la obtención de dichas cuadrículas tomaba una gran cantidad de tiempo por imagen solo se cuenta con las cuadrículas correspondientes a las 5 imágenes. La figura 6-30 muestra estas imágenes con su respectiva etiqueta de clase y etiqueta provista por el conjunto de Nair.

Por motivos de espacio y legibilidad, se mostraran, únicamente, las cuadrículas de activación para las capas 1 y 2 del codificador Resnet-50 de las variantes auto-, semi- y completamente supervisadas (figuras 6-31, 6-32, 6-33, 6-34 y 6-35). Las cuadrículas de activación para las capas 3 y 4 de cada variante se encuentran en el apéndice

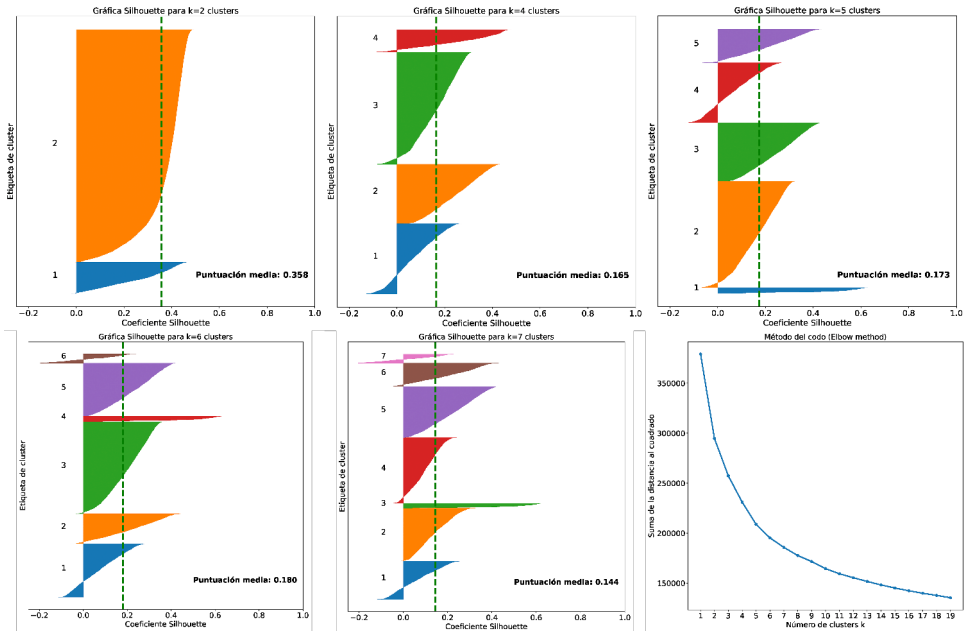


Figura 6-29: Curvas Silhouette y puntación media de Silhouette para superior izquierda: dos grupos, superior medio: 4 grupos, superior derecha: 5 grupos, inferior izquierda: 6 grupos, inferior medio: 7 grupos, inferior derecha: curva de dispersión en función de los grupos (elbow method).

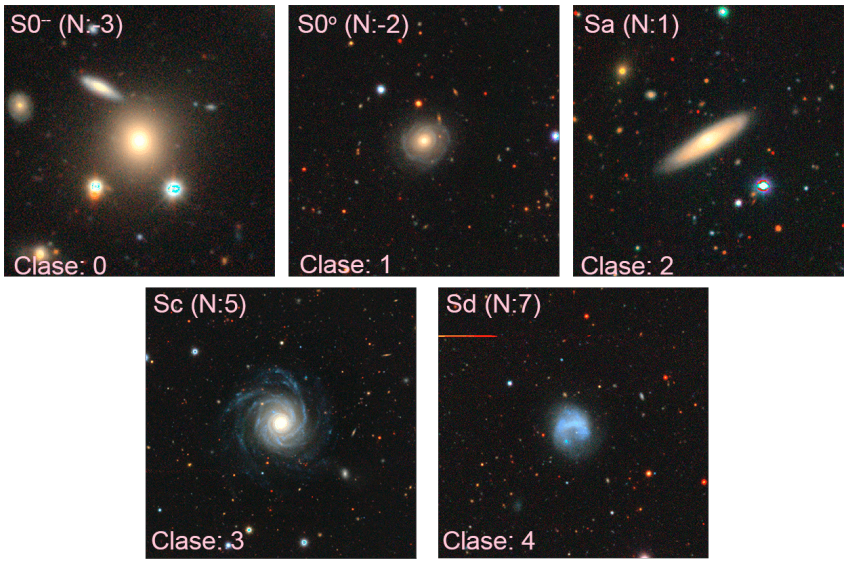


Figura 6-30: Imágenes de referencia para la obtención de las cuadrículas de activación.

A.

Como puede observarse, las cuadrículas de activación para el modelo supervisado distan significativamente de las ofrecidas por los modelos auto- y semi- supervisados. Mientras que para estos dos últimos, son altamente similares entre sí, salvo quizá, dentro de las regiones de interés (galaxias), en donde se observa un ligero cambio en la detección de bordes. Para el caso de las cuadrículas de activación para la capa 2, los métodos auto- y semi- supervisados aparentan preservar los bordes de los objetos originales, lo cual no sucede con el método

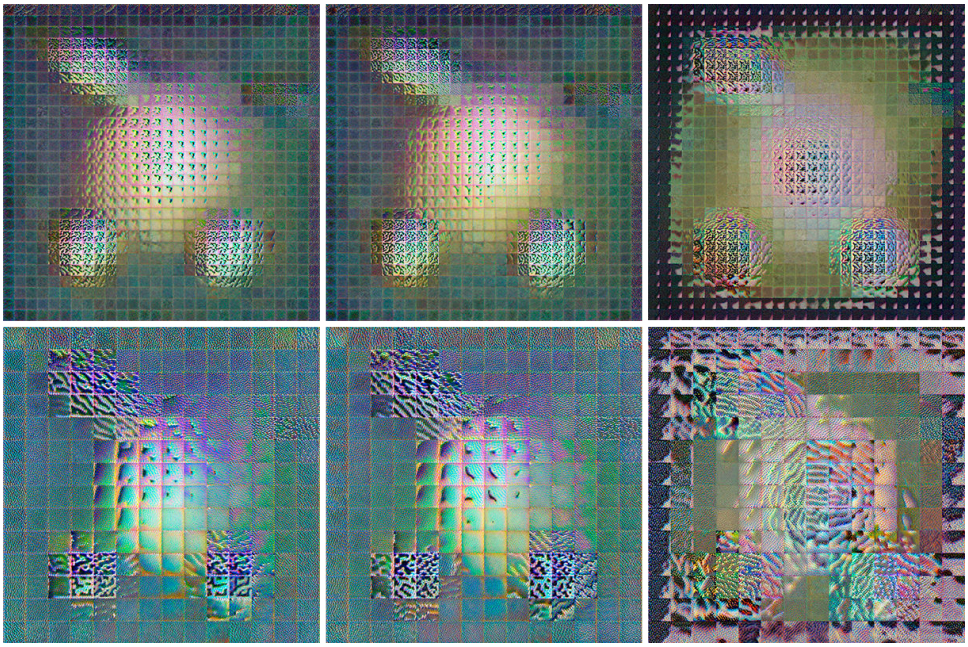


Figura 6-31: Cuadrículas de activación para las capas 1 (superior) y 2 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 0. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

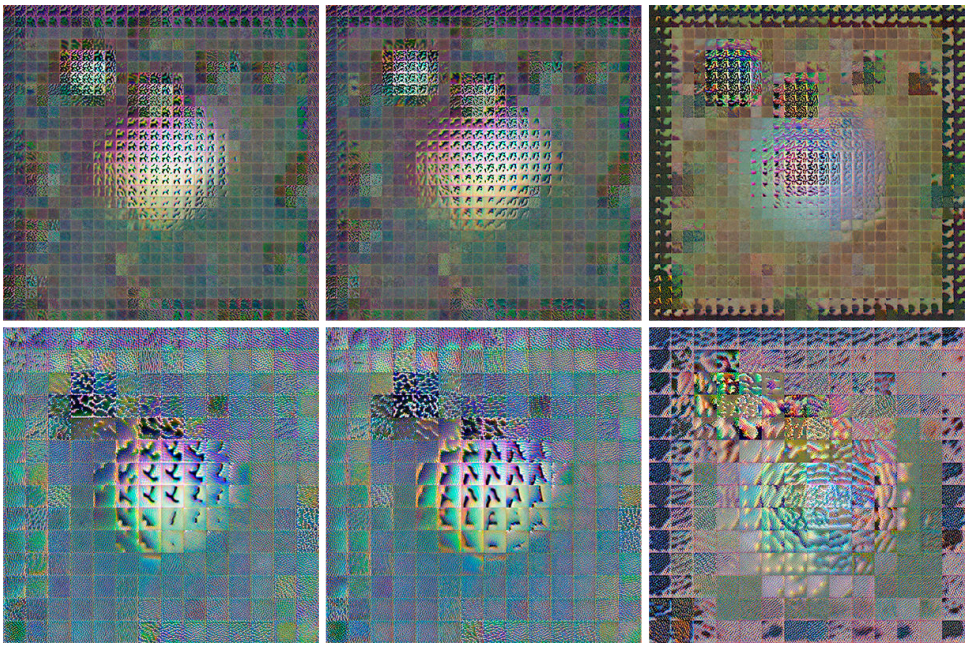


Figura 6-32: Cuadrículas de activación para las capas 1 (superior) y 2 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 1. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

supervisado. Puesto a que el método semisupervisado, es, en esencia, el método autosupervisado, se cree que por esta razón, ambas cuadrículas de activación son altamente similares entre sí.

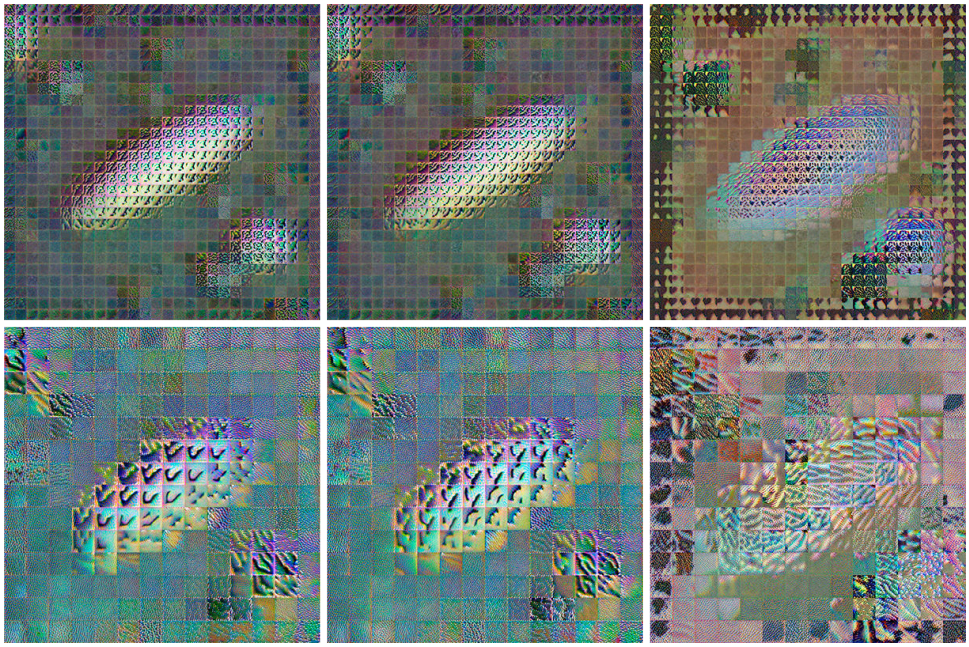


Figura 6-33: Cuadrículas de activación para las capas 1 (superior) y 2 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 2. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

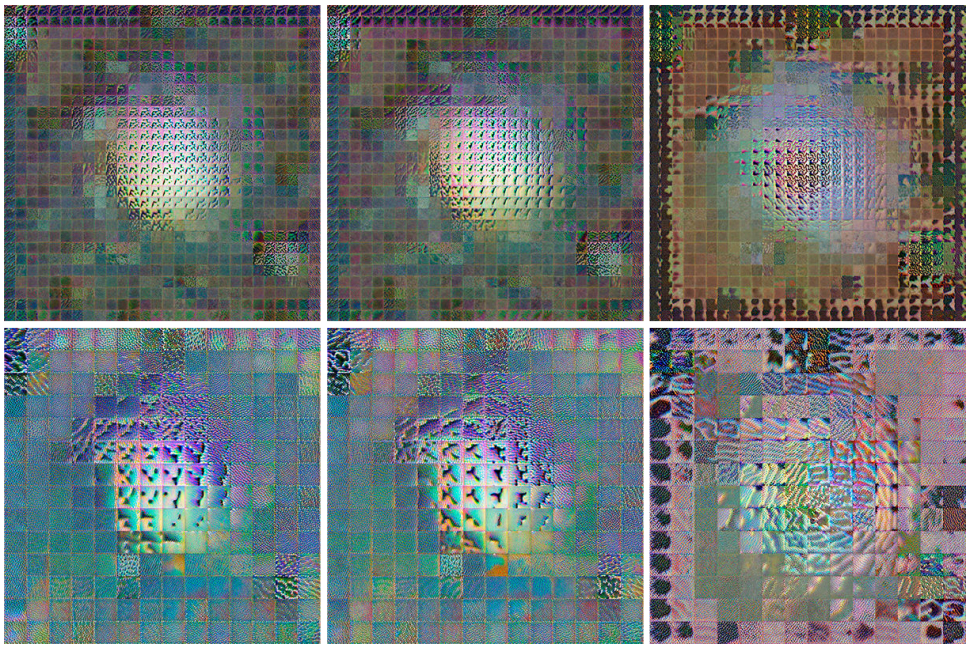


Figura 6-34: Cuadrículas de activación para las capas 1 (superior) y 2 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 3. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

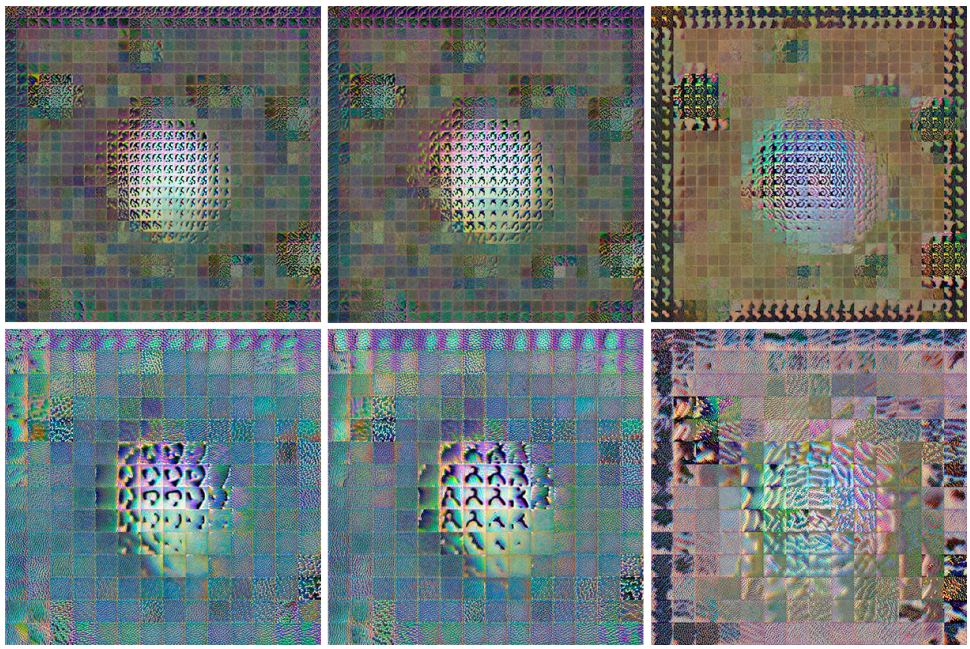


Figura 6-35: Cuadrículas de activación para las capas 1 (superior) y 2 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 4. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

6.7.4. Inversión de características

Las figuras 6-36, 6-37, 6-38, 6-39 y 6-40 muestran las activaciones fuertes para cada capa del modelo autosupervisado a través de cada una de las imágenes de referencia (fig: 6-30). Dado que la interpretación de cada imagen resultante está sujeta a la subjetividad del observador, no se ofrecerá discusión alguna.

A modo de resumen se presentan, en la tabla 6-4, algunas conclusiones más relevantes de cada experimento.

Experimento	Conclusiones
Experimento de referencia	El valor recall más bajo obtenido fue de $50\,\%$ para la clase más desbalanceada (4). Para el ajuste fino con clases balanceadas se obtiene el valor más bajo de recall para la clase 3 de $58\,\%$. Para este último método se observó una ligera disminución en los valores recall, con respecto al ajuste fino con clases balanceadas, en algunas clases, ésto se asocia principalmente a la transición continua entre clases.
Experimentos transformaciones	La transformación extra más importante es el recorte central con proporción de 0.65, recorte y redimensión en el intervalo $[0.2,1.0]$, contraste y saturación en el rango $[0.6,1.4]$, Hue máximo de 0.1 y aplicación de filtros sobel con probabilidad de $30\,\%$. Por otro lado, se encontró que la aplicación de todas las transformaciones posibles produce una especie de saturación la cual perjudica el rendimiento del modelo.
Experimento de referencia modificado	La inclusión de las transformaciones recorte central, filtros de sobel y oscurecimiento, así como la modificación de los rangos en las transformaciones originales resultó en una mejora en la detección de verdaderos positivos para todas las clases, con respecto al experimento de referencia, en, al menos, $1.19\,\%$.
Experimento K-medias	El método con mejores resultados se obtiene al aplicar el algoritmo de K-medias durante todo el entrenamiento. A pesar de esto, para las tres perspectivas estudiadas, se encontró que estos métodos mejoran la detección de verdaderos positivos para las clases más desbalanceadas. Sin embargo, este valor suele decrecer, ínfimamente, para las clases mayoritarias.
Experimentos supervisados y efectividad del método autosupervisado	Los resultados obtenidos para los modelos entrenados de manera completamente supervisada no distan significativamente de los resultados obtenidos por el modelo de SimCLR. Además, dichos resultados son comparables con los reportados en la literatura. Se demostró la efectividad del método autosupervisado.
Experimentos con otros codificadores	El rendimiento obtenido por los codificadores DenseNet-161 y EfficientNetv2 son comparables con respecto al ofrecido por Resnet-50. Sin embargo, durante la fase autosupervisada, DenseNet converge más rápidamente con respecto a su contraparte original.
Visualización	Se infirieron las regiones más relevantes de cada clase gracias al algoritmo SHAP. Se observó que la distribución de clases y real son similares entre sí. Se determinó que la cantidad óptima de clases se encuentra entre 5 y 6. Se observó que las cuadrículas de activación del modelo supervisado difiere con las obtenidas por los modelos auto- y semisupervisados.

Tabla 6-4: Resumen de las conclusiones de cada experimento.

Figura 6-36: Activaciones fuertes para el modelo autosupervisado al emplear la imagen representativa de la clase 0.

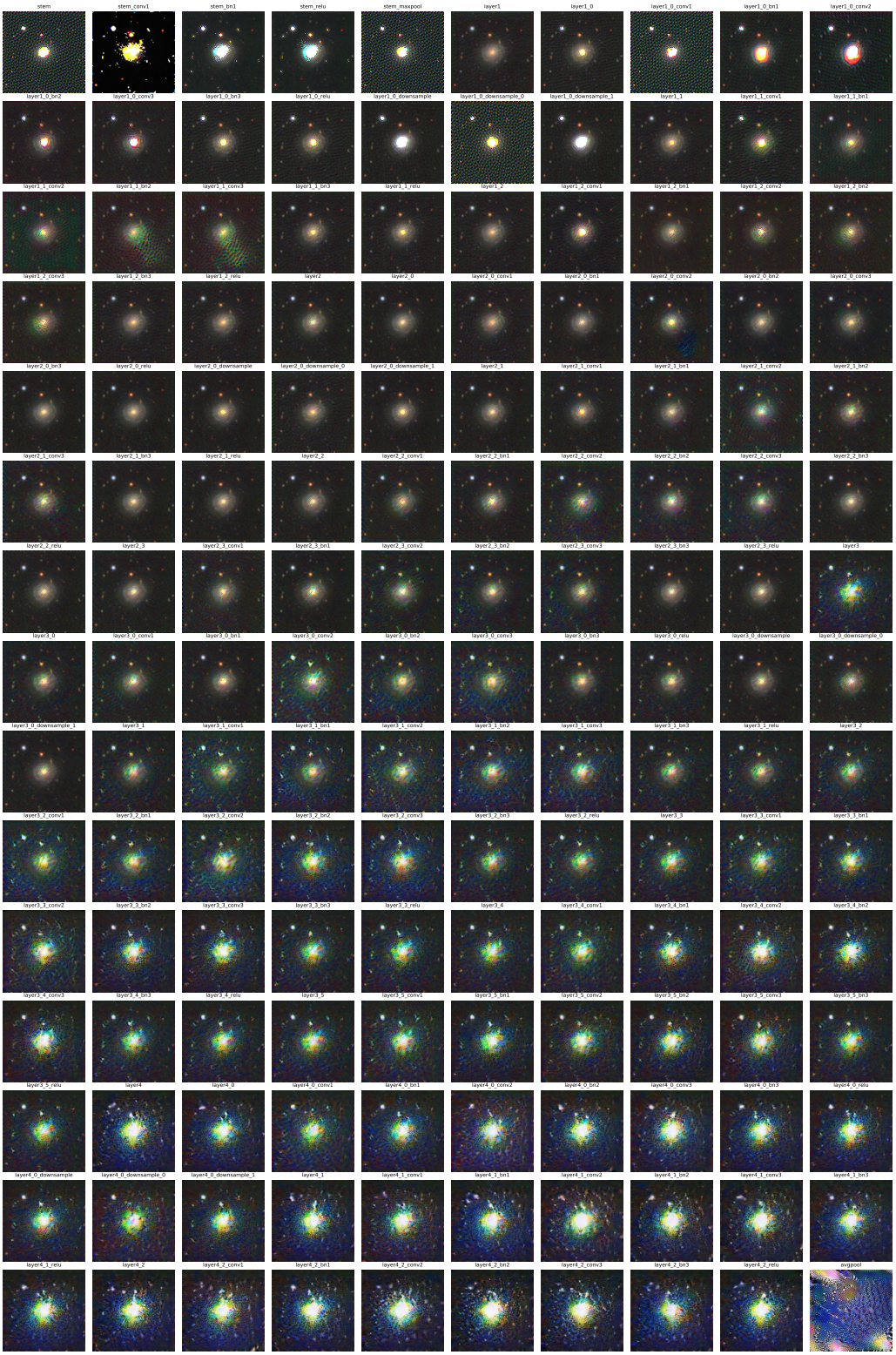


Figura 6-37: Activaciones fuertes para el modelo autosupervisado al emplear la imagen representativa de la clase 1.

Figura 6-38: Activaciones fuertes para el modelo autosupervisado al emplear la imagen representativa de la clase 2.

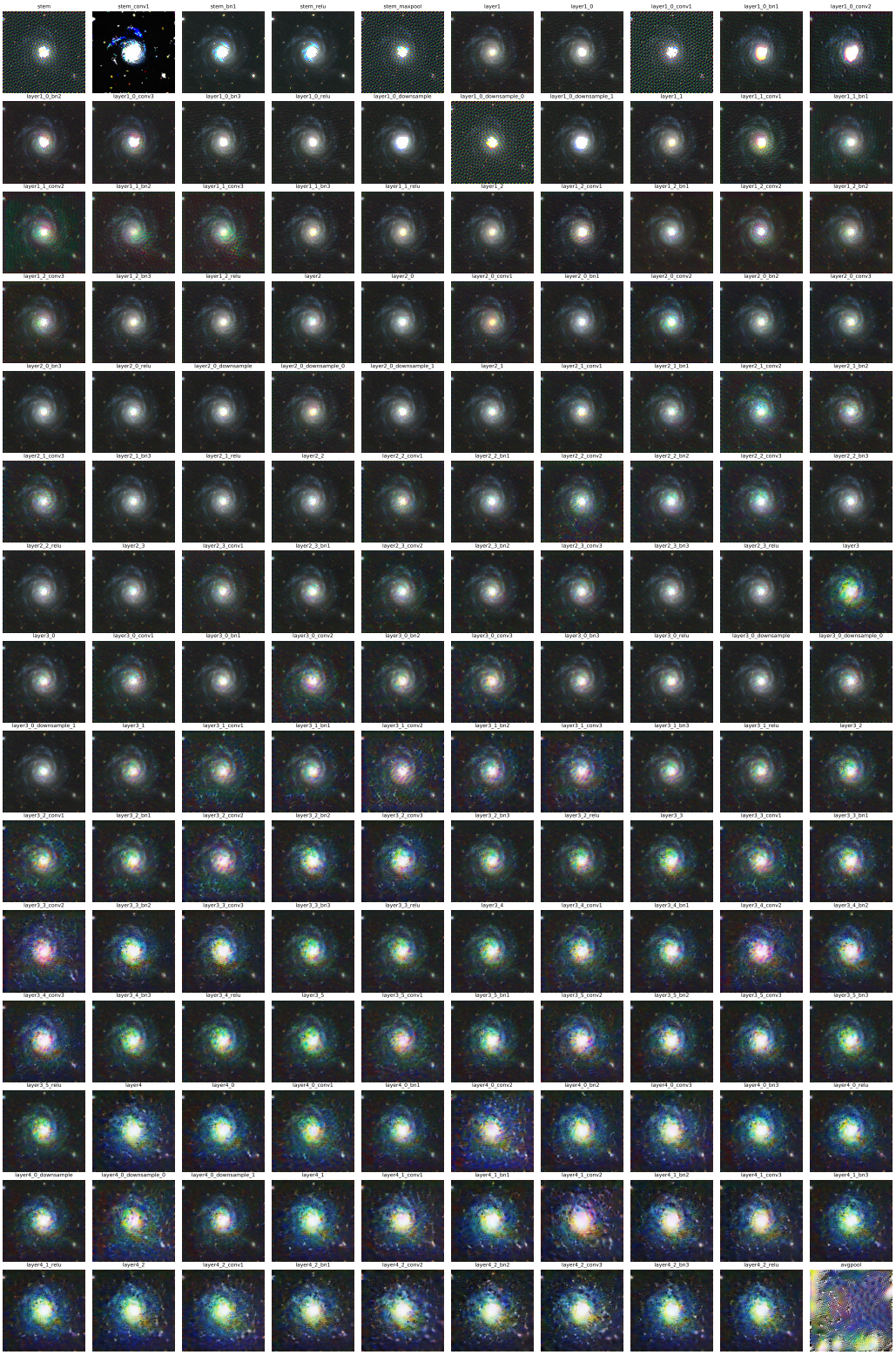


Figura 6-39: Activaciones fuertes para el modelo autosupervisado al emplear la imagen representativa de la clase 3.

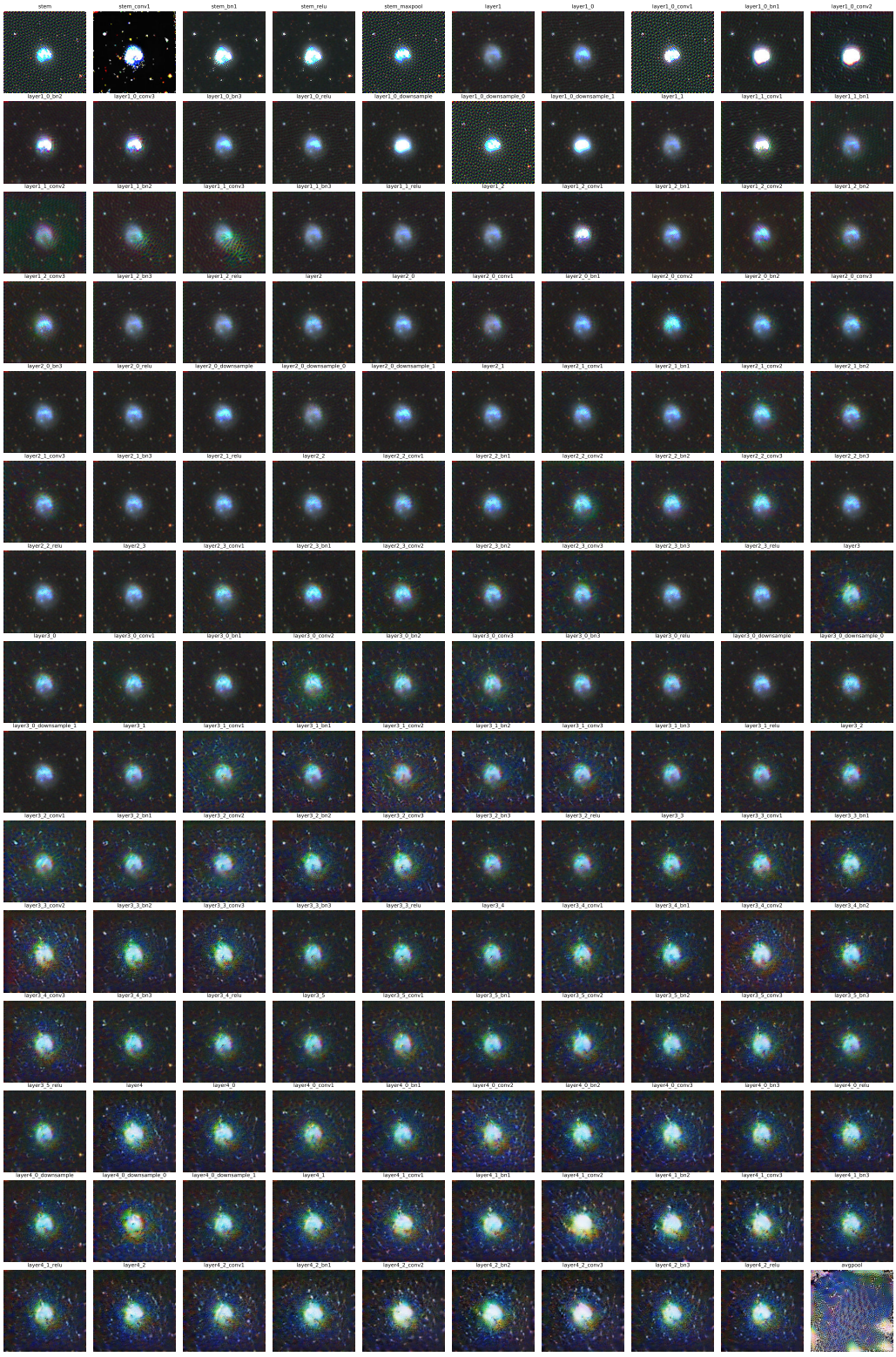


Figura 6-40: Activaciones fuertes para el modelo autosupervisado al emplear la imagen representativa de la clase 4.

Capítulo 7

Conclusiones y trabajo a futuro

A lo largo de este trabajo se presentaron una serie de experimentos, tanto cualitativos como cuantitativos, que nos permitieron estudiar, discutir y comprender algunos hechos relevantes de la tarea de clasificación morfológica de cinco subconjuntos de galaxias. A continuación se presentará una lista donde se resumen los resultados más relevantes obtenidos durante la experimentación. Posterior a dicha lista se enunciarán algunos de los posibles procedimientos y/o caminos a seguir para enriquecer esta investigación.

- Se implementó y logró reproducir con éxito las dos arquitecturas del estado del arte del aprendizaje autosupervisado y semisupervisado para la tarea de clasificación morfológica de cinco grupos de galaxias, empleando dos conjuntos de datos cuyas clases eran debalanceadas. Este es uno de los aportes, considerados, más significativos, pues este problema en particular ha sido muy poco estudiado desde esta perspectiva.

- El estudio del preprocesamiento de las imágenes durante la fase autosupervisada, permitió descubrir que, para las imágenes de índole astronómica, las transformaciones extra más relevantes son el recorte central, oscurecimiento, así como filtros de Sobel. Se cree que éstas permiten regularizar el conjunto de datos, pues se pone más atención a las regiones de interés y reduce la cantidad de ruido producida por luminosidad. De esta forma, al implementar dichas transformaciones, se encontró que la proporción de verdaderos positivos aumentó, para las clases desbalanceadas, mientras que para las clases mayoritarias, este valor se mantuvo o se acortó insignificativamente, con respecto a las transformaciones originales.

- Para combatir el desbalance de clases y, por ende, mejorar la proporción de verdaderos positivos para las clases más desbalanceadas, se implementó la arquitectura del estado del arte autosupervisada junto con el algoritmo de K-medias, el cual generaba pseudoetiquetas de acuerdo a la distribución espacial de los vectores de representación de cada imagen para, a través de las mismas, generar un conjunto de entrenamiento, por cada época, pseudobalanceado. Para este análisis se realizaron tres experimentos, en donde, se aplicaba el algoritmo de agrupamiento a lo largo de todo, primera y segunda mitad del entrenamiento. Los resultados obtenidos para estos tres fueron satisfactorios, logrando incrementar hasta $9.2\,\%$ la proporción de verdaderos positivos en la clase más desbalanceada.

- Se demostró, experimentalmente, que el método autosupervisado aporta información general del conjunto de entrenamiento, la cual es ajustada durante el método semisupervisado.

- Se realizó una comparación directa entre los métodos semisupervisado y completamente supervisado, obteniéndose resultados similares entre ambos, sin embargo, como es de esperarse, el método completamente supervisado logra superar, ligeramente, al semisupervisado. Esta comparación no se centró en decidir cuál método es mejor, si no más bien, inferir el comportamiento de las etiquetas sobre los modelos supervisados y de esta manera compararlos con los resultados de los estudios que emplean dicho método. De esta comparación, se infiere que el conjunto de etiquetas, empleado durante todos los experimentos, puede mejorarse.

- Se estudió el comportamiento de las métricas de evaluación al cambiar el codificador original (Resnet-50) por codificadores más actuales como lo son DenseNet-161 y EfficientNetv2. Los resultados obtenidos no muestran una mejoría significativa a favor de algún codificador en particular, sin embargo, se probó que la arquitectura DenseNet-161 converge aproximadamente dos veces más rápido que su contraparte original.

- Se obtuvieron las imágenes SHAP que muestran las regiones de decisión más relevantes para cada clase aprendidas por el modelo semisupervisado a través del algoritmo SHapley Additive exPlanations para un subconjunto de 50 imágenes consideradas representativas de cada clase. De estas regiones de interés, se infiere que, aparentemente, los atributos más relevantes para cada clase son: el centro de galaxias o regiones luminosas para las etiquetas 0 y 1. La región total comprendida por la galaxia, con brazos en caso de existir, para las etiquetas 2 y 3 y la región central de cada galaxia, así como los al rededores de la misma para la etiqueta 4.

- Gracias a la naturaleza de la visualización del algoritmo SHAP y el punto anterior, es posible extraer una idea más general acerca de las características más relevantes de cada clase y, de esta manera, usarlas a favor de mejorar el rendimiento de los modelos a través de un nuevo conjunto de transformaciones que logre incluir y/o resaltar dichas características. Como se pudo observar, durante los experimentos de transformaciones, la inclusión del recorte central y filtros de Sobel mejora la correcta identificación en la mayoría de clases comparada con el conjunto de transformaciones originales propuestas en SimCLRv2.

- Gracias a la distribución de pseudoclases y clases reales fue posible identificar que el modelo semisupervisado ofrece una distribución de pseudoclases similar a la real, con lo cual, fue posible inferir la distribución de clases en el conjunto no etiquetado.

- Se comprobó que la cantidad de grupos o clases óptima, para el conjunto de datos, es de dos. Sin embargo, al descartar este valor, se encontró que el valor subóptimo se encuentra entre 5 y 6. Para dicha comprobación se obtuvieron las curvas de Silhouette y elbow.

- Se obtuvieron las activaciones fuertes, así como las cuadriculas de activación por cada capa del codificador Resnet-50 perteneciente a los modelos autosupervisado, semisupervisado y supervisado. Si bien, la interpretación de estos resultados es compleja, éstos ofrecen un panorama general del tratamiento que cada modelo da a cada imagen.

7.1. Trabajo a futuro

Con base en los resultados obtenidos, las discusiones generadas y los problemas inherentes dentro del problema de la clasificación morfológica de galaxias se propone seguir las siguientes lineas de investigación:

- Un aspecto del que se estuvo consciente durante el desarrollo de este trabajo fue la calidad de las imágenes presentes en el conjunto de datos no etiquetado, existiendo algunas para las cuales se presenta una especie de aberración cromática muy evidente o la luminosidad de ciertos objetos es de tal forma que el objeto de interés perdía alguna o varías características relevantes, produciendo así imágenes ruidosas sin ningún tipo de relevancia. De esta forma, se propone realizar un método automático que elimine dichas imágenes. Este método podría considerar los histogramas de color de cada imagen para así uniformizar el conjunto de datos. Puesto a que este conjunto no cuenta con etiquetas, también podría considerarse cambiarlo completamente, es decir, extraer un subconjunto de datos de los repositorios más actuales, a modo de que la calidad de las imágenes sea superior.

- Para combatir el desbalance de clases, se demostró que el algoritmo de K-medias mejora la cantidad de verdaderos positivos, aunque para nuestro caso se limitaron la cantidad de imágenes por clase a tres mil, se cree que aumentar, este valor umbral podría mejorar los resultados. Por otro lado, también podrían implementarse otros modelos no supervisados basados en agrupamiento, para inferir y pseudobalancear el conjunto de datos. Otra idea que no fue desarrollada en este trabajo, pero se cree que podría colaborar

en el problema del desbalance de clases, es la implementación de un término extra de regularización sobre la función de pérdida NT-Xent.

- El cambio de codificador probó incrementar el tiempo de convergencia durante la fase autosupervisada, sin embargo, es necesario estudiar esto más formalmente, a través de una serie de experimentos que contemplen el cambio del rendimiento del modelo en función de la cantidad de épocas de preentrenamiento. Por otro lado, el desarrollo de los experimentos desbalanceados y balanceados de la arquitectura, más compleja, DenseNet-201, complementaría los resultados e hipótesis ofrecidas en esta sección.
- Se cree que la discrepancia entre los resultados obtenidos por los modelos supervisados y los reportados por diversos estudios se debe a la definición de las cinco clases a la que nos sujetamos. De esta forma, se considera que la distribución de clases debe ser mejor estudiada, a modo de que la transición continua entre clases no sea tan notoria.
- A pesar de que la mejora relativa obtenida por el nuevo conjunto de transformaciones no fue significativa, es posible estudiar una serie de transformaciones extra no exploradas en este trabajo, de tal forma que se consideren los resultados visuales obtenidos por el algoritmo SHAP.

Apéndice A

Resultados complementarios

A.1. Experimento de referencia

A.1.1. Codificador como extractor de características

La figura A-1 muestra las métricas de clasificación para el modelo cuyo ajuste fino modificó únicamente los pesos de la capa de clasificación usando los conjuntos de entrenamiento y validación desbalanceados. Como puede verse, el modelo no logra obtener el $50\,\%$ de exhaustividad para la clase 4, la cual contiene la menor cantidad de etiquetas. Este hecho podría explicar dicho valor, sin embargo, obsérvese que el valor de exhaustividad para la clase mayoritaria (clase 2) es significativamente menor al presentado por la segunda clase mayoritaria (clase 0), además, a partir de la matriz de confusión (Fig. A-1c), se sabe que, el modelo clasifica erróneamente una cantidad significativa de imágenes que corresponden a clases contiguas. De esta manera, surge la posibilidad de que, además del problema de desbalance de clases, la clasificación de galaxias enfrenta un problema con respecto a la definición o subclasificación, el cual puede ser atribuido a una transición continua entre cada una de las clases.

Al comparar el área bajo la curva de las curvas ROC-AUC y Precision-Recall de las clases 1 y 4 con respecto a las otras tres (Figs. A-1a y A-1b), puede detectarse un rendimiento que consideramos bajo y es atribuido principalmente al desbalance de clases. Así mismo, el valor $F1$ para estas dos clases minoritarias no superan el $53\,\%$.

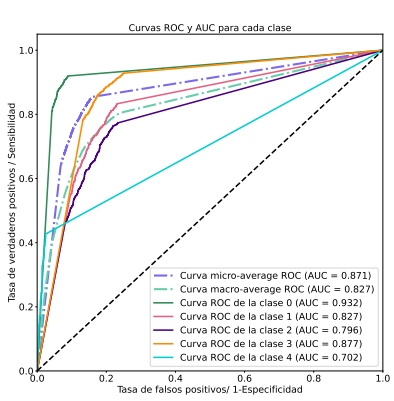
Las métricas de clasificación para el modelo usado como extractor de características y los conjuntos de datos balanceados se muestran en la figura A-2.

El uso de los conjuntos balanceados permite a la red una mejor detección de los verdaderos positivos. Sin embargo, al igual que en los casos en los cuales se emplea el codificador como inicialización, la clase 3 sufre una caída en esta proporción. A diferencia de los casos anteriores, se observa este mismo hecho para la clase 2 (la segunda con menor cantidad de etiquetas). A este fenómeno, lo asociaremos principalmente, al diseño e implementación de las transformaciones, pues al observar los resultados del experimento de referencia modificado bajo las mismas condiciones (ver fig. A-4) este fenómeno no se produce.

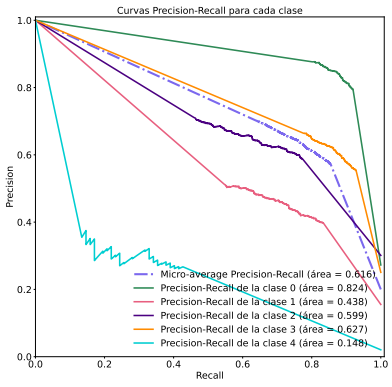
A.2. Experimento de referencia modificado

A.2.1. Codificador como extractor de características

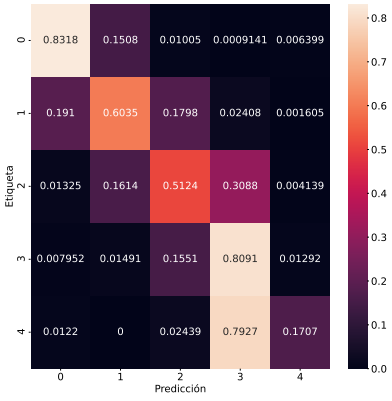
La figura A-3 muestra las métricas de clasificación obtenidas al entrenar únicamnete la capa supervisada, empleando los conjuntos de entrenamiento y validación desbalanceados, mientras que la figura A-4 corresponden a las métricas de clasificación empleando ambos conjuntos balanceados.



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	86.34 %	83.18 %	84.73 %
1	50.07 %	60.35 %	54.73 %
2	68.78 %	51.12 %	58.73 %
3	64.20 %	80.91 %	71.59 %
4	35.00 %	17.07 %	22.95 %
macro avg	60.88 %	58.55 %	58.55 %
weighted avg	68.82 %	68.10 %	67.69 %
Exactitud	68.10 %		

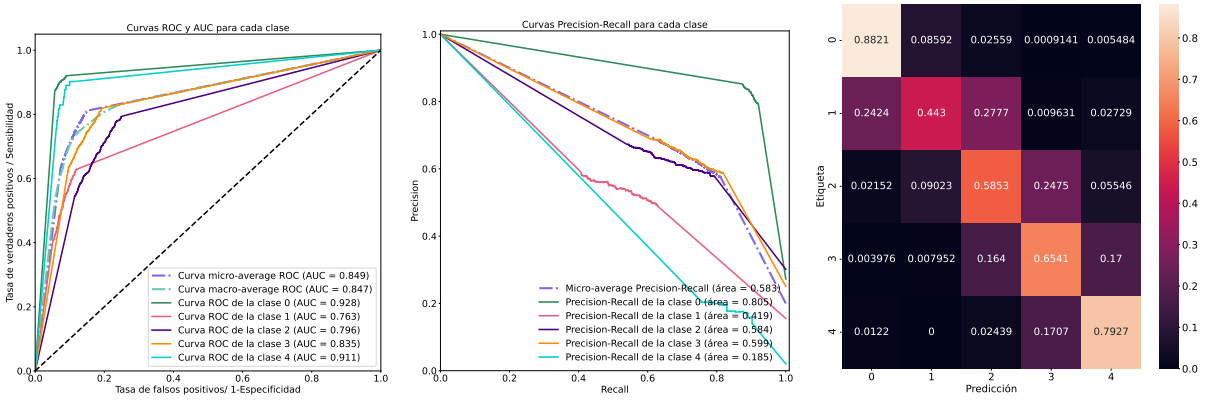
(d) Reporte de clasificación

Figura A-1: Métricas de evaluación obtenidas para el experimento de referencia usando el codificador como extractor de características.

Ahora bien, note que la matriz de confusión, para el caso de la red extractora de características, debe ser tratada de manera especial; sí se recuerda que en esta red, los únicos pesos que pueden ser actualizados son aquellos que pertenecen a la capa de clasificación, entonces, es factible esperar que la matriz de confusión brinde indicios sobre la calidad de las representaciones aprendidas durante el método autosupervisado. Notemos que para los casos balanceados y desbalanceados (figs. A-4c y A-3c) la clase 0 resulta con un valor de, al menos, 80 %, lo cual indica que, para esta clase, las representaciones pueden ser consideradas de calidad. Por otro lado, debido a la variación significativa entre los valores por clase de estos dos indicadores, no puede ofrecerse un argumento similar al anterior. Un hecho importante de mencionar es que, para ambos casos, el sistema presenta una clasificación errónea y significativa para las clases contiguas. Siendo el principal motivo de esto, la naturaleza continua de clases en la morfología de galaxias.

A.3. Experimentos transformaciones

Las tablas A-1, A-2, A-3 y A-4 muestran la pérdida y exactitud en los conjuntos de válidación entrenamiento y prueba para cada uno de los experimentos planteados en el capítulo 5, sección 5.3. Como ya se mencionó, se eligió aquellos experimentos para los cuales el promedio de la exactitud en prueba sea mayor, así como la desviación estándar sea pequeña. La tabla 6-1 resume dichos valores para cada uno de los experimentos, en donde las columnas *Exactitud Contrastiva* y *Pérdida Contrastiva*, corresponden a los valores obtenidos al final del entrenamiento autosupervisado, es por esta razón que dichos valores se repiten en cada tabla. Por otro lado, las columnas restantes, excepto la última, corresponden a valores para los cuales el valor de exactitud en validación fue mayor.



(a) Curva ROC-AUC

(b) Curva Precision-Recall

(c) Matriz de confusión

Clase	Precision	Recall	F1
0	84.13 %	88.21 %	86.12 %
1	56.67 %	44.30 %	49.73 %
2	65.77 %	58.53 %	61.94 %
3	67.28 %	65.41 %	66.33 %
4	19.94 %	79.27 %	31.86 %
macro avg	58.76 %	67.14 %	59.20 %
weighted avg	68.80 %	66.56 %	67.12 %
Exactitud	66.56 %		

(d) Reporte de clasificación

Figura A-2: Métricas de evaluación obtenidas para el experimento de referencia usando el codificador como extractor de características y conjuntos de datos balanceados.

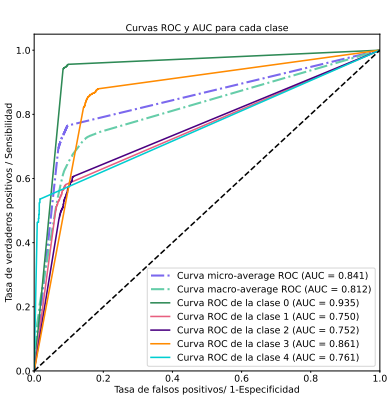
Experimento (tasa de aprendizaje 5×10^{-3})	Exactitud Contrastiva[%]	Pérdida Contrastiva[u.a.]	Exactitud Train[%]	Pérdida Train[u.a]	Exactitud Validación[%]
E1	95.61	0.8662	76.95	1.276	66.02
E2	90.54	1.329	64.84	4.778	67.96
E3	87.32	1.632	72.66	1.900	67.31
E4	85.03	1.847	72.26	0.7542	67.26
E5	87.79	1.566	80.86	0.4837	38.21
E6	88.29	1.592	73.43	1.180	66.62
E7	56.26	4.565	64.06	3.261	66.21
E8	99.48	0.2888	82.81	0.3846	67.22
E9	67.64	3.468	78.91	0.8598	68.06
E10	83.52	2.004	74.22	1.177	66.07
E11	82.79	2.046	76.17	1.081	67.71
E12	85.35	1.799	69.53	2.335	68.41

Tabla A-1: Resultados de los experimentos propuestos en la tabla 5-3 con una tasa de aprendizaje de 5×10^{-3} durante el ajuste fino.

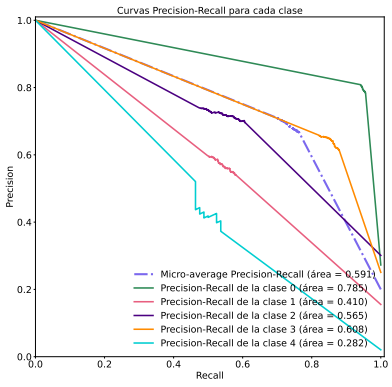
A.4. Visualización

A.4.1. SHapley Additive exPlanations

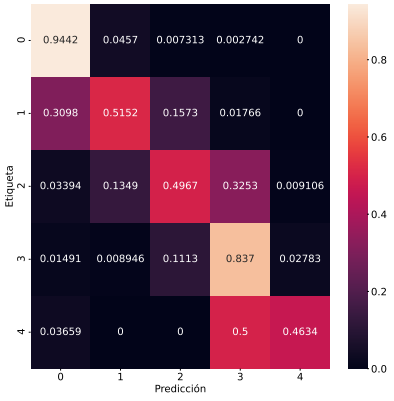
La figura A-5 muestra el conjunto total de imágenes originales transformadas por el algoritmo SHAP y el codificador semisupervisado. Mientras que, las figuras A-6, A-7, A-8, A-9 y A-10 corresponden a las imágenes



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	80.39 %	94.42 %	86.84 %
1	59.12 %	51.52 %	55.06 %
2	73.35 %	49.67 %	59.23 %
3	65.27 %	83.70 %	73.34 %
4	49.35 %	46.34 %	47.80 %
macro avg	65.50 %	65.13 %	64.46 %
weighted avg	70.54 %	70.62 %	69.42 %
Exactitud	70.62 %		

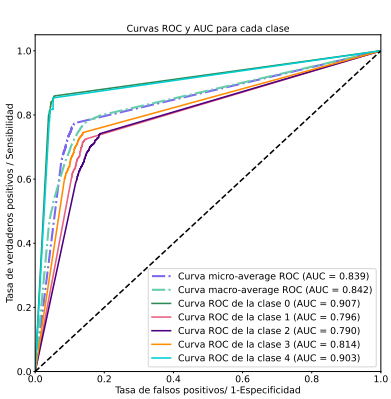
(d) Reporte de clasificación

Figura A-3: Métricas de evaluación obtenidas para el experimento de referencia modificado usando el codificador como extractor de características y conjuntos de entrenamiento y validación desbalanceados.

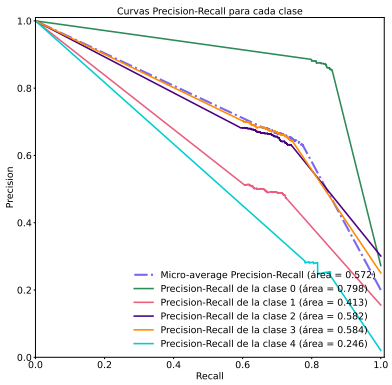
Experimento (tasa de aprendizaje 2×10^{-3})	Exactitud Contrastiva[%]	Pérdida Contrastiva[u.a.]	Exactitud Train[%]	Pérdida Train[u.a]	Exactitud Validación[%]
E1	95.61	0.8662	61.72	4.975	65.51
E2	90.54	1.329	77.73	0.6330	68.26
E3	87.32	1.632	70.70	1.370	69.86
E4	85.03	1.847	73.44	0.5856	68.41
E5	87.79	1.566	83.20	0.4635	68.56
E6	88.29	1.592	74.21	1.144	67.46
E7	56.26	4.565	68.36	1.899	66.31
E8	99.48	0.2888	77.34	0.5264	67.07
E9	67.64	3.468	70.31	1.209	68.26
E10	83.52	2.004	72.66	1.323	67.42
E11	82.79	2.046	67.58	2.034	67.46
E12	85.35	1.799	69.92	1.402	68.81

Tabla A-2: Resultados de los experimentos propuestos en la tabla 5-3 con una tasa de aprendizaje de 2×10^{-3} durante el ajuste fino.

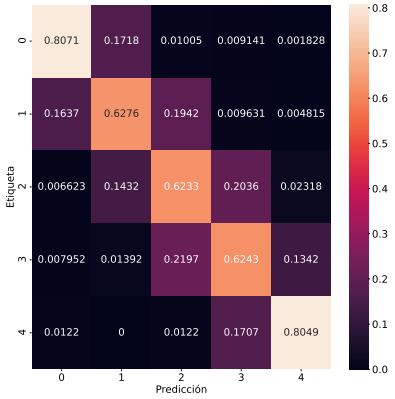
SHAP complementarias para cada clase.



(a) Curva ROC-AUC



(b) Curva Precision-Recall



(c) Matriz de confusión

Clase	Precision	Recall	F1
0	88.12 %	80.71 %	84.26 %
1	51.04 %	62.76 %	56.30 %
2	68.02 %	62.33 %	65.05 %
3	69.47 %	62.42 %	65.76 %
4	28.20 %	80.49 %	41.77 %
macro avg	60.97 %	69.74 %	62.63 %
weighted avg	70.42 %	67.80 %	68.63 %
Exactitud	67.80 %		

(d) Reporte de clasificación

Figura A-4: Métricas de evaluación obtenidas para el experimento de referencia modificado usando el codificador como extractor de características y conjuntos de entrenamiento y validación balanceados.

Experimento (tasa de aprendizaje 1×10^{-3})	Exactitud Contrastiva[%]	Pérdida Contrastiva[u.a.]	Exactitud Train[%]	Pérdida Train[u.a]	Exactitud Validación[%]
E1	95.61	0.8662	70.70	0.8564	65.62
E2	90.54	1.329	78.52	0.5274	68.41
E3	87.32	1.632	77.73	0.6121	68.16
E4	85.03	1.847	67.97	0.7626	68.41
E5	87.79	1.566	77.73	0.5225	69.16
E6	88.29	1.592	73.83	0.7063	67.86
E7	56.26	4.565	62.50	1.824	65.12
E8	99.48	0.2888	73.05	0.6486	67.12
E9	67.64	3.468	71.88	0.8553	67.07
E10	83.52	2.004	65.23	1.267	67.52
E11	82.79	2.046	67.19	1.319	66.77
E12	85.35	1.799	72.66	0.6648	68.31

Tabla A-3: Resultados de los experimentos propuestos en la tabla 5-3 con una tasa de aprendizaje de 1×10^{-3} durante el ajuste fino.

A.4.2. Cuadrículas de activación

Las cuadrículas de activación correspondientes a las capas 3 y 4 del codificador Resnet-50 para cada uno de los tres modelos auto-, semi- y completamente supervisados, a través de las imágenes de referencia de la figura 6-30, se presentan en las figuras A-11, A-12, A-13, A-14 y A-15. De dichas figuras, puede apreciarse que las capas posteriores no preservan la estructura morfológica de las imágenes originales, si no que cada uno de los filtros

Experimento (tasa de aprendizaje 7×10^{-4})	Exactitud Contrastiva[%]	Pérdida Contrastiva[u.a.]	Exactitud Train[%]	Pérdida Train[u.a]	Exactitud Validación[%]
E1	95.61	0.8662	72.66	0.6506	65.97
E2	90.54	1.329	75.78	0.6212	68.76
E3	87.32	1.632	74.22	0.6205	68.01
E4	85.03	1.847	73.44	0.6603	68.26
E5	87.79	1.566	76.95	0.5378	69.26
E6	88.29	1.592	75.78	0.6863	68.16
E7	56.26	4.565	60.94	1.671	65.32
E8	99.48	0.2888	77.34	0.5502	66.82
E9	67.64	3.468	75.39	0.6191	67.52
E10	83.52	2.004	67.97	0.9056	67.17
E11	82.79	2.046	70.70	0.6948	67.02
E12	85.35	1.799	74.22	0.6464	68.66

Tabla A-4: Resultados de los experimentos propuestos en la tabla 5-3 con una tasa de aprendizaje de 7×10^{-4} durante el ajuste fino.

extrae para cada una de las imágenes patrones complejos, que a simple vista son difíciles de interpretar.

A.4.3. Inversión de características

Las activaciones fuertes correspondientes al modelo semisupervisado no serán presentadas, pues, las imágenes resultantes son altamente similares entre ésta y las presentadas en las figuras 6-36, 6-37, 6-38, 6-39 y 6-40. En contraparte, las figuras A-16, A-17, A-18, A-19 y A-20 corresponden a las activaciones fuertes del modelo completamente supervisado para cada una de las imágenes de referencia (fig. 6-30).

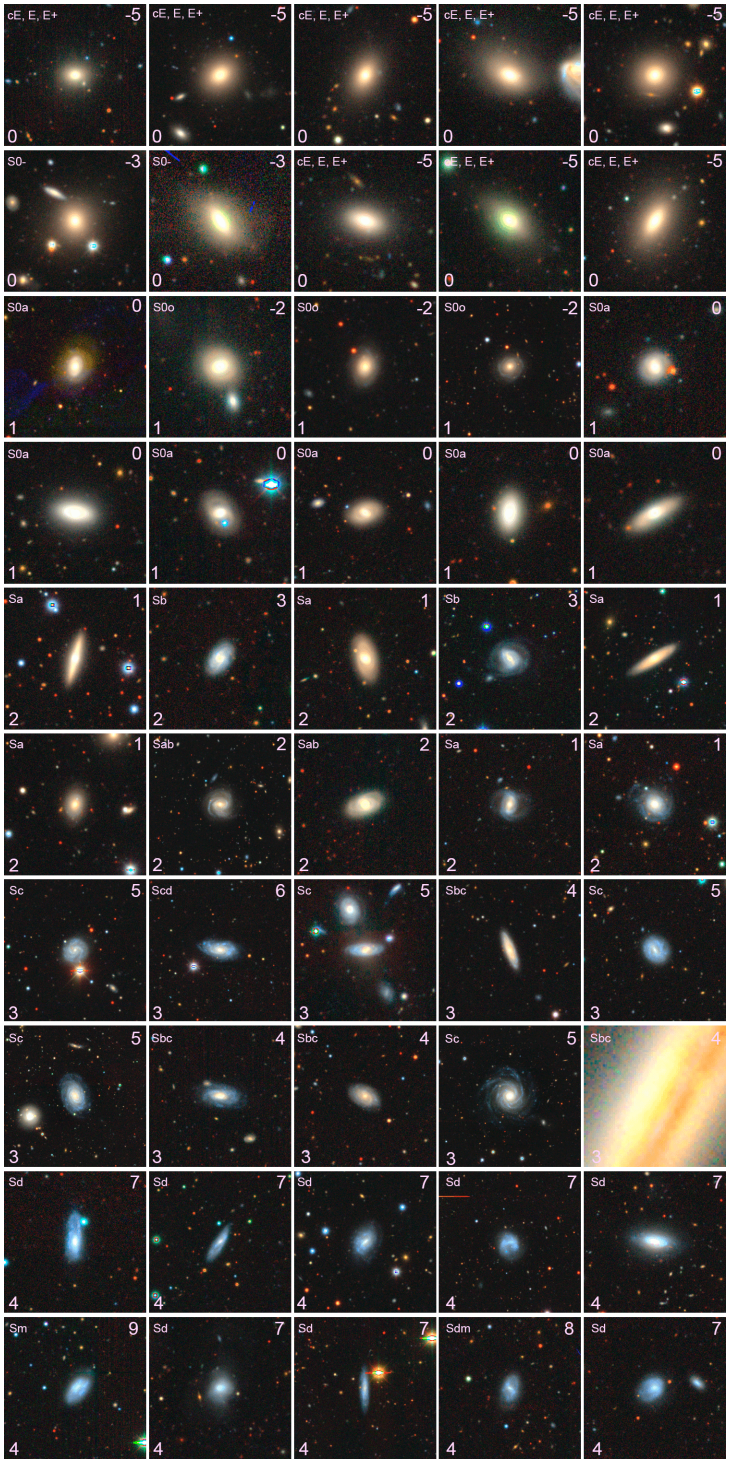


Figura A-5: Imágenes de referencia para el algoritmo SHAP. Superior derecha: clasificación Nair, inferior izquierda: subclasificación empleada para entrenar todos los modelos.

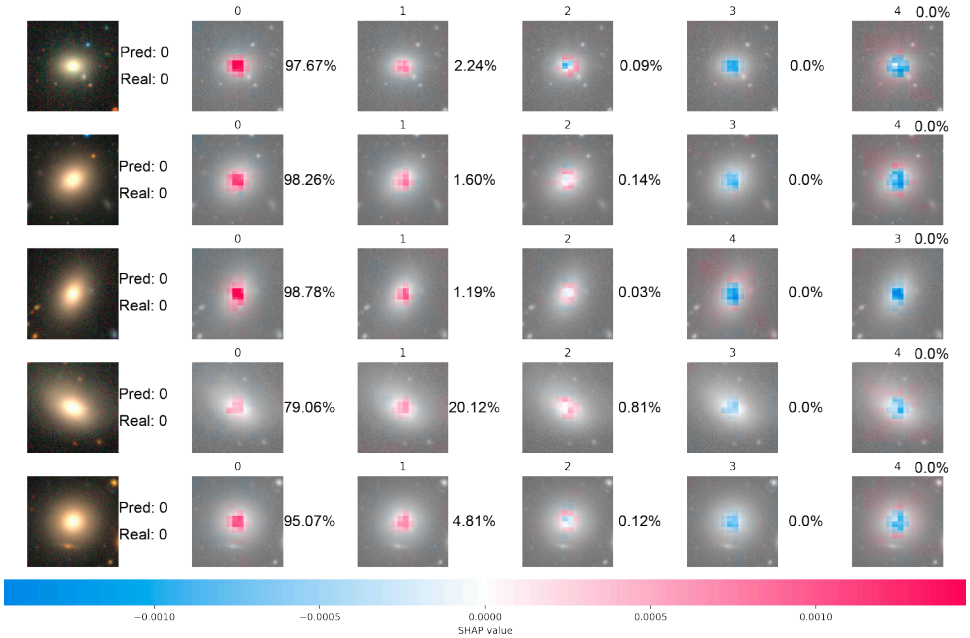


Figura A-6: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 0 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

Figura A-7: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 1 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

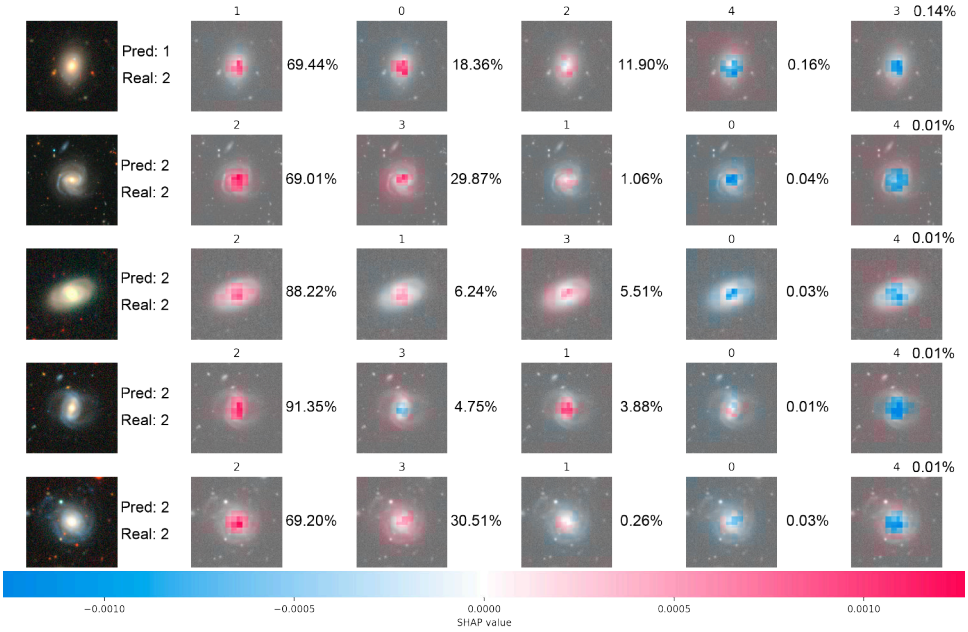


Figura A-8: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 2 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

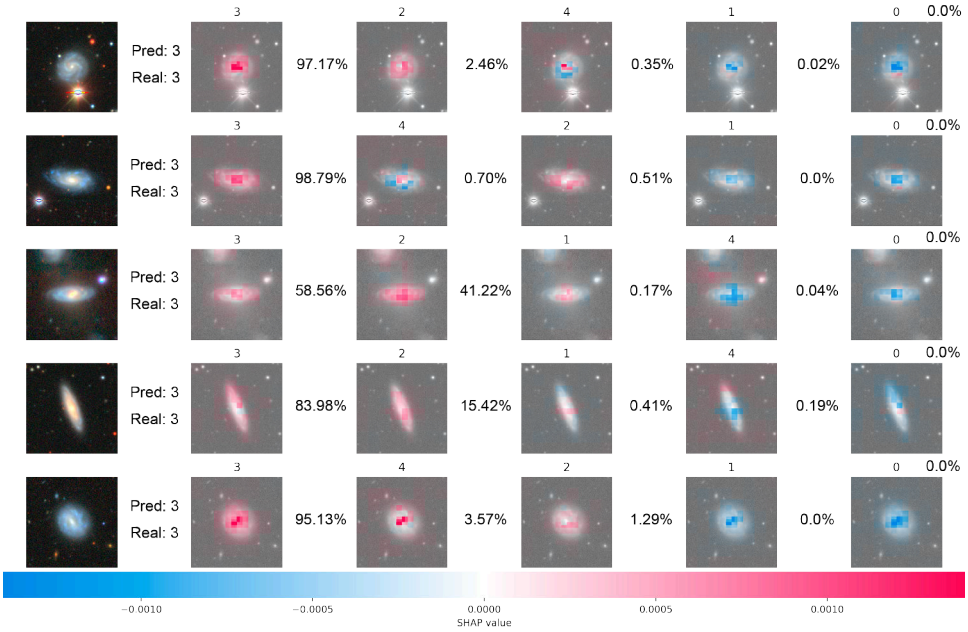


Figura A-9: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 3 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

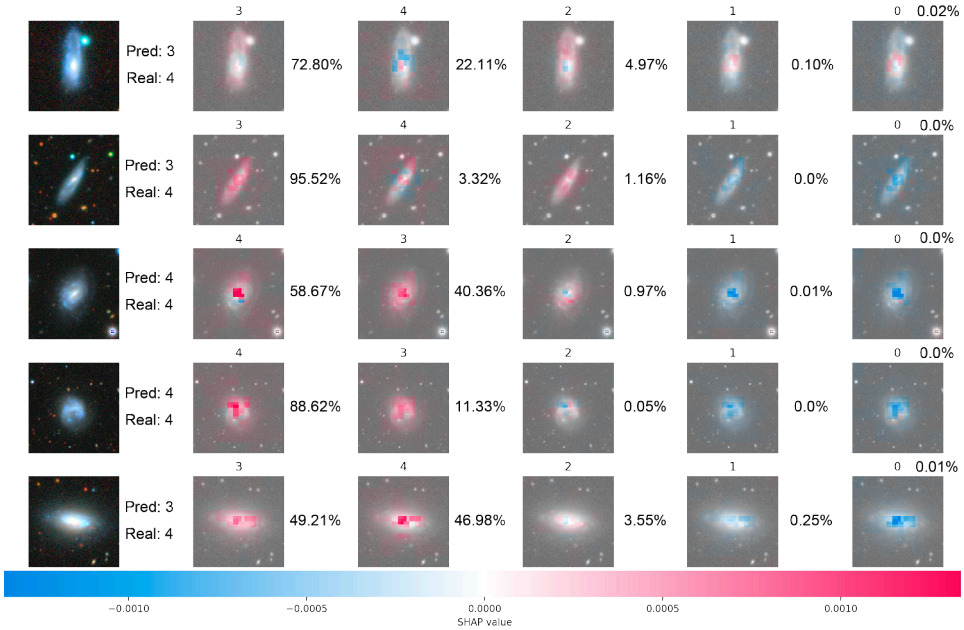


Figura A-10: Imágenes SHAP para un subconjunto de 5 imágenes pertenecientes a la clase 4 del conjunto Nair. El valor en la parte superior corresponde a una de las cinco clases disponibles, a su lado derecho, se muestra la probabilidad de su correspondiente etiqueta.

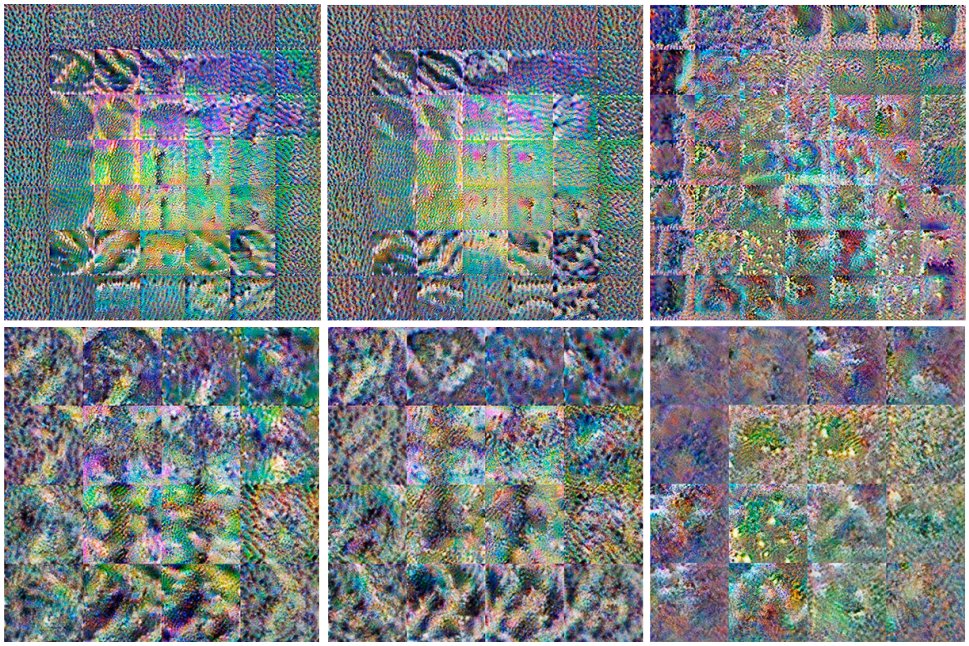


Figura A-11: Cuadrículas de activación para las capas 3 (superior) y 4 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 0. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

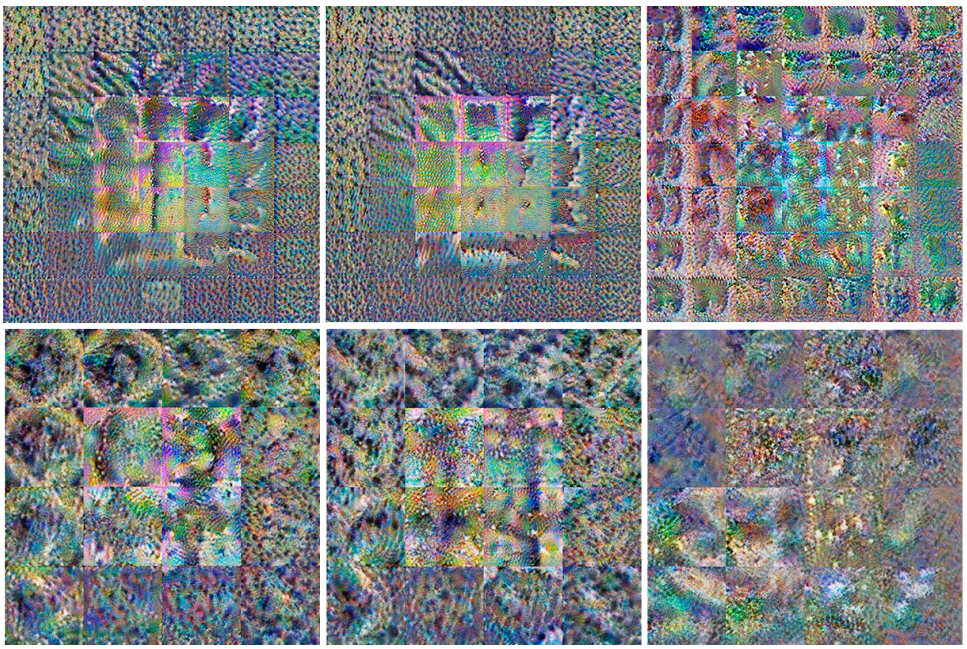


Figura A-12: Cuadrículas de activación para las capas 3 (superior) y 4 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 1. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

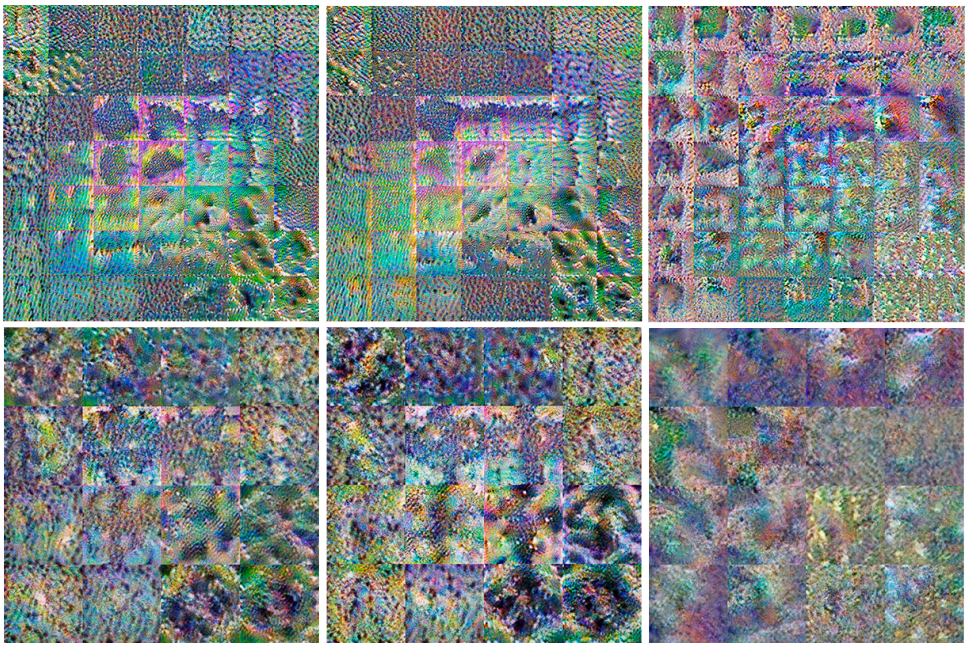


Figura A-13: Cuadrículas de activación para las capas 3 (superior) y 4 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 2. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

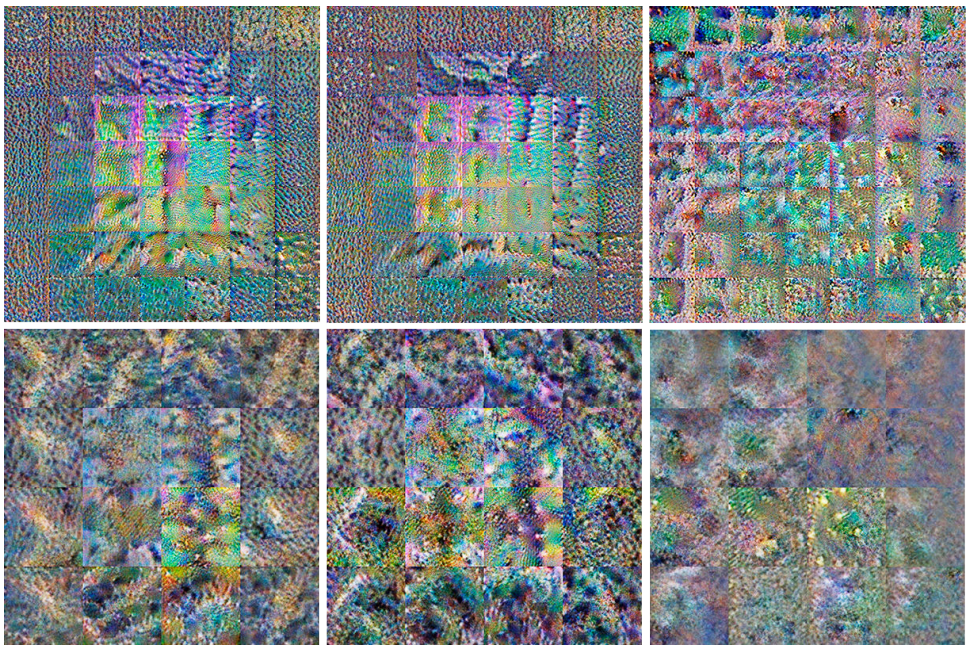


Figura A-14: Cuadrículas de activación para las capas 3 (superior) y 4 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 3. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

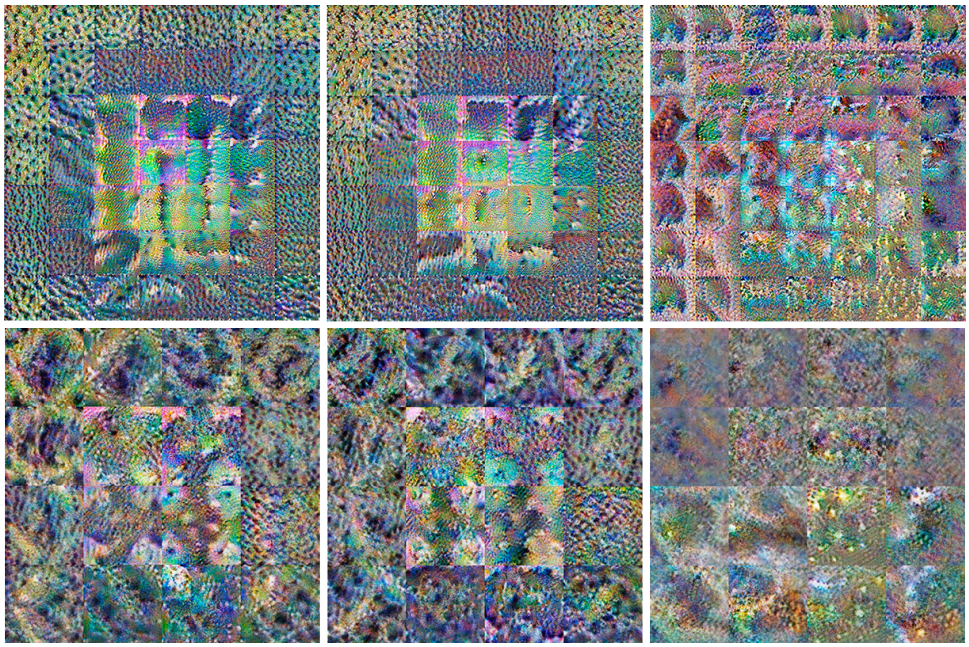


Figura A-15: Cuadrículas de activación para las capas 3 (superior) y 4 (inferior) de la arquitectura Resnet-50 para la imagen representativa de la clase 4. Izquierda: Autosupervisado, medio: Semisupervisado, derecha: Supervisado.

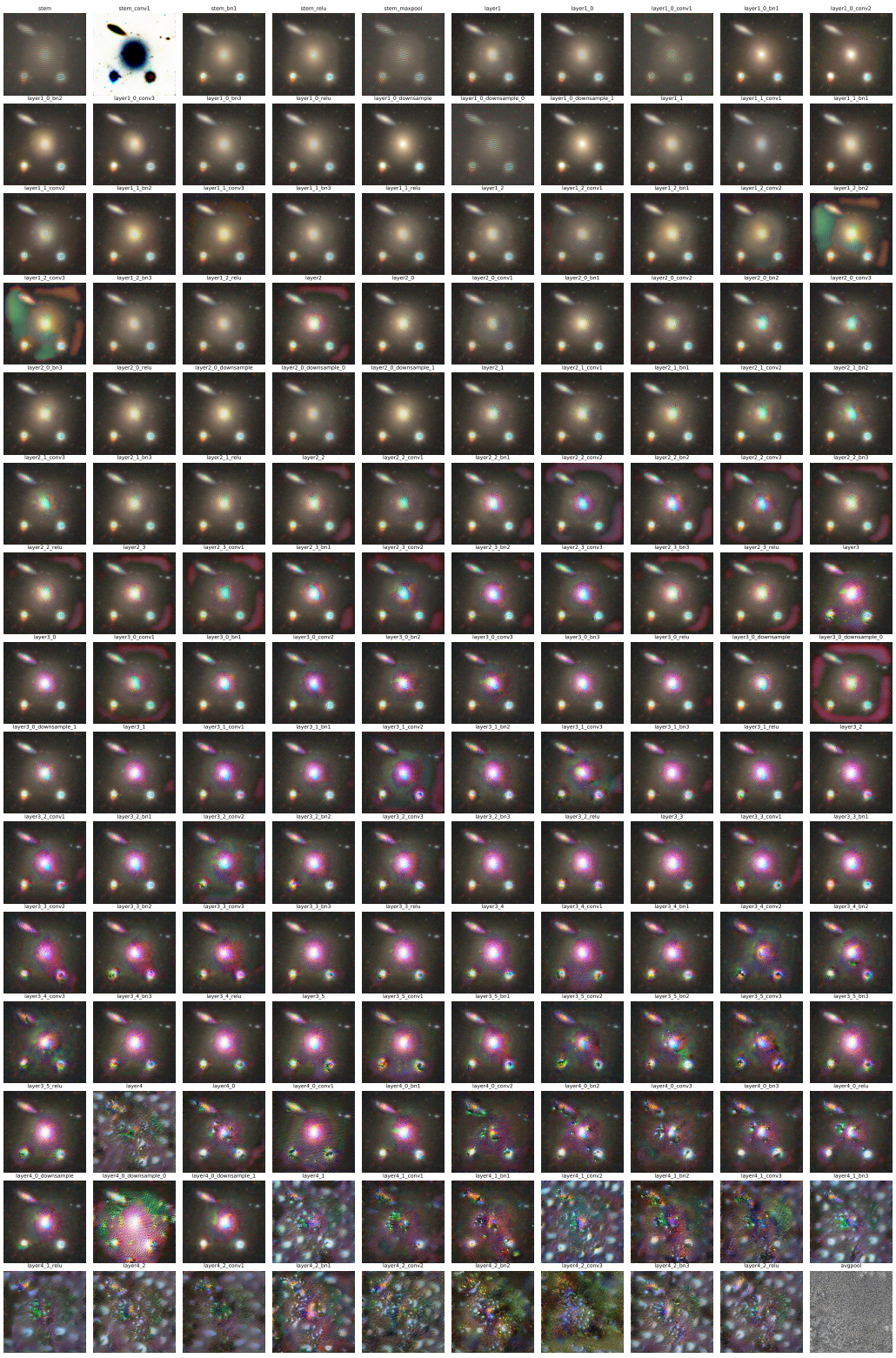


Figura A-16: Activaciones fuertes para el modelo supervisado al emplear la imagen representativa de la clase 0.

Figura A-17: Activaciones fuertes para el modelo supervisado al emplear la imagen representativa de la clase 1.

Figura A-18: Activaciones fuertes para el modelo supervisado al emplear la imagen representativa de la clase 2.

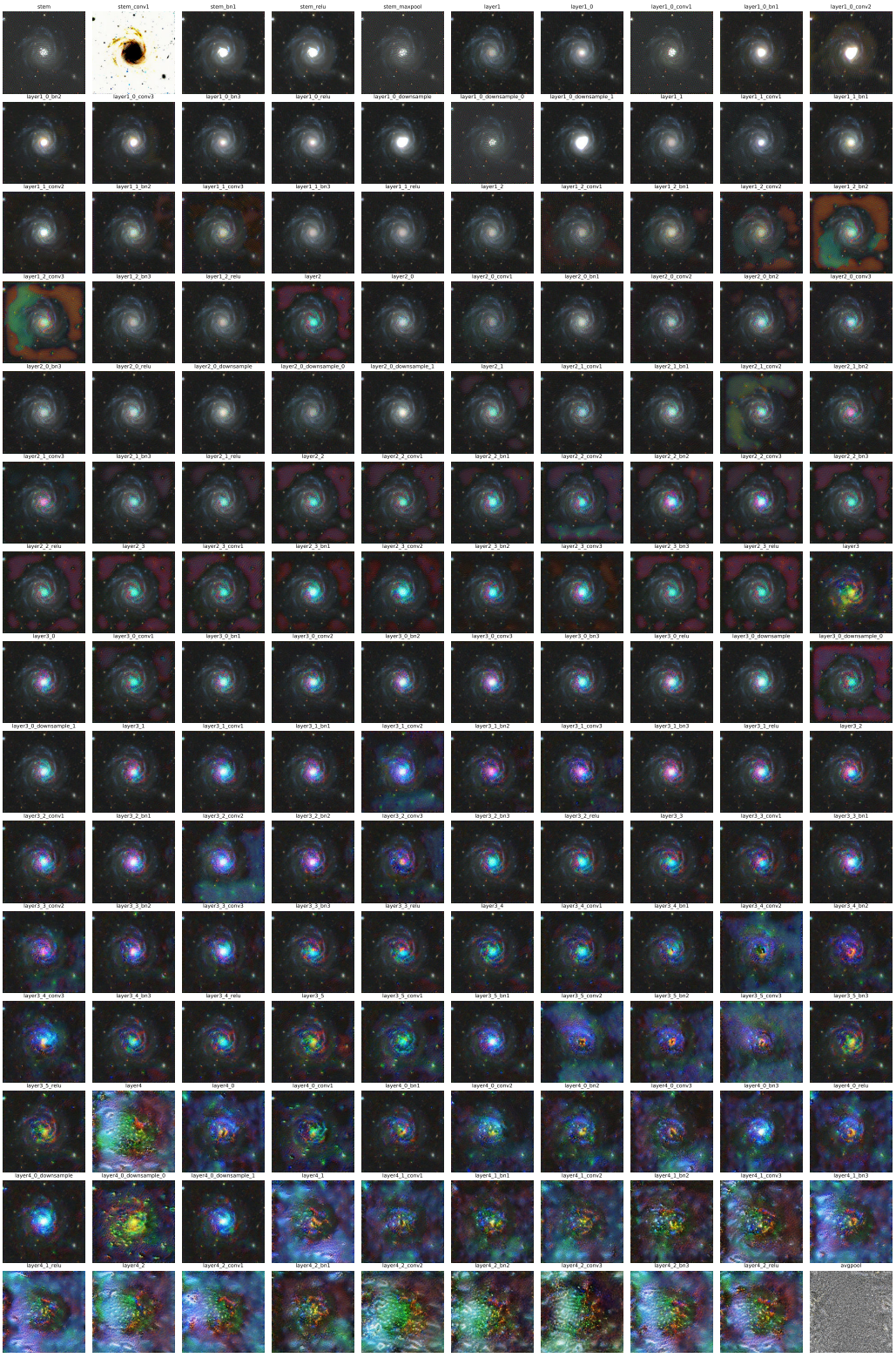


Figura A-19: Activaciones fuertes para el modelo supervisado al emplear la imagen representativa de la clase 3.

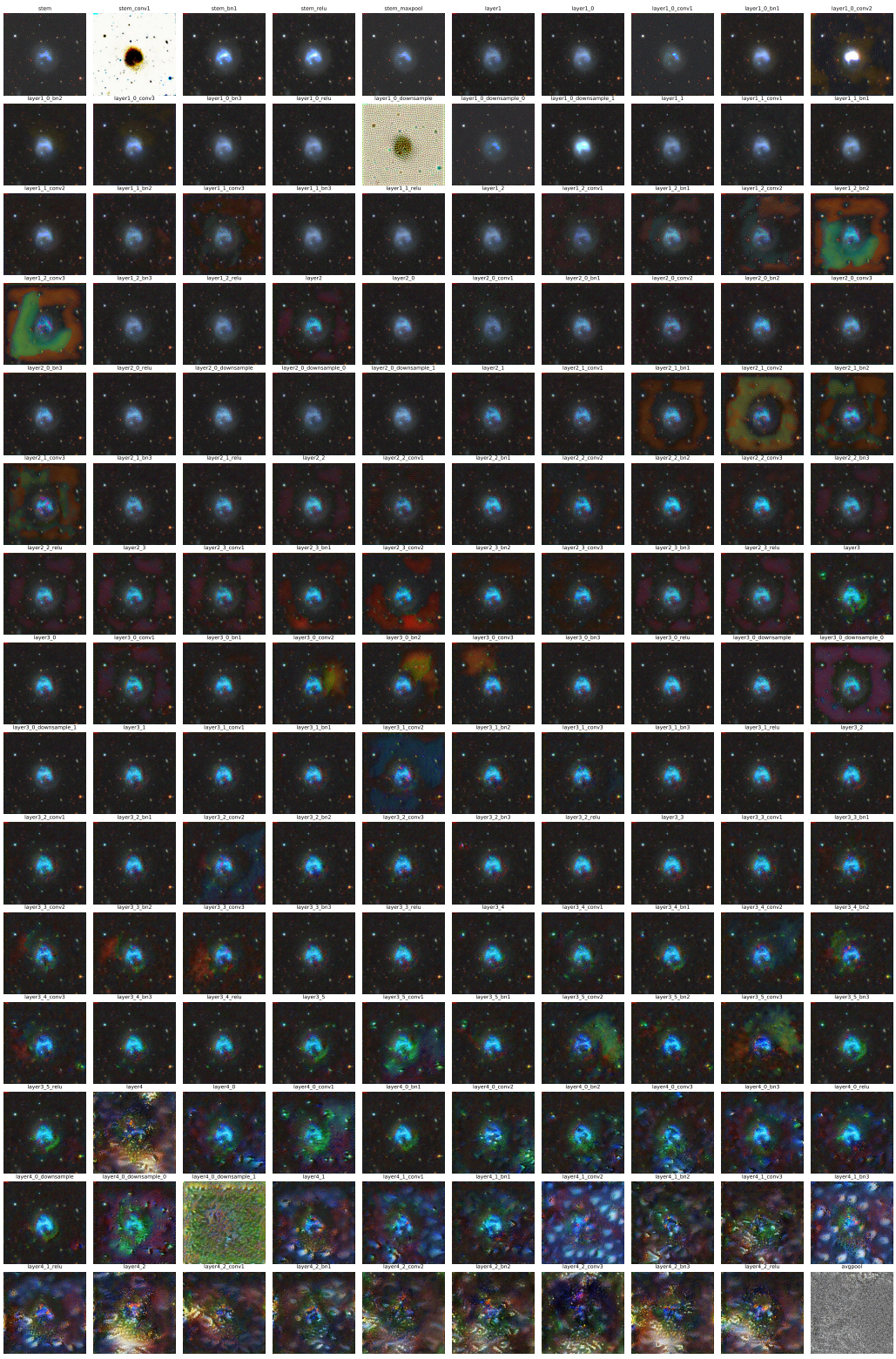


Figura A-20: Activaciones fuertes para el modelo supervisado al emplear la imagen representativa de la clase 4.

Bibliografía

[1] C. de Toro y Llaca, *Astronomía: Historia y Calendario*, vol. 194. Instituto de astronomía y geodesia, 1999.

[2] E. P. Hubble, "Extragalactic nebulae.," *The Astrophysical Journal*, vol. 64, p. 321, 1926.

[3] P. H. Barchi, R. R. de Carvalho, R. R. Rosa, R. A. Sautter, M. Soares-Santos, B. A. Marques, E. Clua, T. S. Gonçalves, C. de Sá-Freitas, and T. C. Moura, "Machine and deep learning applied to galaxy morphology - a comparative study," *Astronomy and Computing*, vol. 30, 2020.

[4] S. G. Djorgovski, A. Mahabal, A. Drake, M. Graham, and C. Donalek, "Sky surveys," 2013.

[5] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, M. J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg, "Galaxy zoo 1: Data release of morphological classifications for nearly 900 000 galaxies," *Monthly Notices of the Royal Astronomical Society*, vol. 410, pp. 166–178, 1 2011.

[6] K. N. Abazajian, J. K. Adelman-Mccarthy, M. A. Agüeros, S. S. Allam, C. A. Prieto, D. An, K. S. Anderson, S. F. Anderson, J. Annis, N. A. Bahcall, C. A. Bailer-Jones, J. C. Barentine, B. A. Bassett, A. C. Becker, T. C. Beers, E. F. Bell, V. Belokurov, A. A. Berlind, E. F. Berman, M. Bernardi, S. J. Bickerton, D. Bizyaev, J. P. Blakeslee, M. R. Blanton, J. J. Bochanski, W. N. Boroski, H. J. Brewington, J. Brinchmann, J. Brinkmann, R. J. Brunner, T. Budavri, L. N. Carey, S. Carliles, M. A. Carr, F. J. Castander, D. Cinabro, A. J. Connolly, I. Csabai, C. E. Cunha, P. C. Czarapata, J. R. Davenport, E. D. Haas, B. Dilday, M. Doi, D. J. Eisenstein, M. L. Evans, N. W. Evans, X. Fan, S. D. Friedman, J. A. Frieman, M. Fukugita, B. T. Gänsicke, E. Gates, B. Gillespie, G. Gilmore, B. Gonzalez, C. F. Gonzalez, E. K. Grebel, J. E. Gunn, Z. Györy, P. B. Hall, P. Harding, F. H. Harris, M. Harvanek, S. L. Hawley, J. J. Hayes, T. M. Heckman, J. S. Hendry, G. S. Hennessy, R. B. Hindsley, J. Hoblitt, C. J. Hogan, D. W. Hogg, J. A. Holtzman, J. B. Hyde, S. I. Ichikawa, T. Ichikawa, M. Im, E. Ivezić, S. Jester, L. Jiang, J. A. Johnson, A. M. Jorgensen, M. Jurić, S. M. Kent, R. Kessler, S. J. Kleinman, G. R. Knapp, K. Konishi, R. G. Kron, J. Krzesinski, N. Kuropatkin, H. Lampeitl, S. Lebedeva, M. G. Lee, Y. S. Lee, R. F. Leger, S. Lépine, N. Li, M. Lima, H. Lin, D. C. Long, C. P. Loomis, J. Loveday, R. H. Lupton, E. Magnier, O. Malanushenko, V. Malanushenko, R. Mandelbaum, B. Margon, J. P. Marriner, D. Martínez-Delgado, T. Matsubara, P. M. McGehee, T. A. McKay, A. Meiksin, H. L. Morrison, F. Mullally, J. A. Munn, T. Murphy, T. Nash, A. Nebot, E. H. Neilsen, H. J. Newberg, P. R. Newman, R. C. Nichol, T. Nicinski, M. Nieto-Santisteban, A. Nitta, S. Okamura, D. J. Oravetz, J. P. Ostriker, R. Owen, N. Padmanabhan, K. Pan, C. Park, G. Pauls, J. Peoples, W. J. Percival, J. R. Pier, A. C. Pope, D. Pourbaix, P. A. Price, N. Purger, T. Quinn, M. J. Raddick, P. R. Fiorentin, G. T. Richards, M. W. Richmond, A. G. Riess, H. W. Rix, C. M. Rockosi, M. Sako, D. J. Schlegel, D. P. Schneider, R. D. Scholz, M. R. Schreiber, A. D. Schwope, U. Seljak, B. Sesar, E. Sheldon, K. Shimasaku, V. C. Sibley, A. E. Simmons, T. Sivarani, J. A. Smith, M. C. Smith, V. Smolić, S. A. Snedden, A. Stebbins, M. Steinmetz, C. Stoughton, M. A. Strauss, M. Subbarao, Y. Suto, A. S. Szalay, I. Szapudi, P. Szkody, M. Tanaka, M. Tegmark, L. F. Teodoro, A. R. Thakar, C. A. Tremonti, D. L. Tucker, A. Uomoto, D. E. V. Berk, J. Vandenberg, S. Vidrih, M. S. Vogeley, W. Voges, N. P. Vogt, Y. Wadadekar, S. Watters, D. H. Weinberg, A. A. West, S. D. White, B. C. Wilhite, A. C. Wonders, B. Yanny, D. R. Yocum, D. G. York, I. Zehavi, S. Zibetti, and D. B. Zucker, "The seventh data release of the sloan digital sky survey," *Astrophysical Journal, Supplement Series*, vol. 182, pp. 543–558, 2009.

[7] K. W. Willett, C. J. Lintott, S. P. Bamford, K. L. Masters, B. D. Simmons, K. R. Casteels, E. M. Edmondson, L. F. Fortson, S. Kaviraj, W. C. Keel, T. Melvin, R. C. Nichol, M. J. Raddick, K. Schawinski, R. J. Simpson, R. A. Skibba, A. M. Smith, and D. Thomas, "Galaxy zoo 2: Detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey," *Monthly Notices of the Royal Astronomical Society*, vol. 435, pp. 2835–2860, 11 2013.

[8] T. Y. Cheng, C. J. Conselice, A. Aragón-Salamanca, N. Li, A. F. Bluck, W. G. Hartley, J. Annis, D. Brooks, P. Doel, J. García-Bellido, D. J. James, K. Kuehn, N. Kuropatkin, M. Smith, F. Sobreira, and G. Tarle, "Optimising automatic morphological classification of galaxies with machine learning and deep learning using dark energy survey imaging," 2019.

[9] M. C. Storrie-Lombardi, O. Lahav, L. Sodre, and L. J. Storrie-Lombardi, "Morphological classification of galaxies by artificial neural networks," *Monthly Notices of the Royal Astronomical Society*, vol. 259, pp. 8P–12P, 11 1992.

[10] J. D. L. Calleja and O. Fuentes, "Machine learning and image analysis for morphological galaxy classification," *Monthly Notices of the Royal Astronomical Society*, vol. 349, pp. 87–93, 3 2004.

[11] A. Naim, O. Lahav, L. Sodre, and M. C. Storrie-Lombardi, "Automated morphological classification of apm galaxies by supervised artificial neural networks," *Monthly Notices of the Royal Astronomical Society*, vol. 275, pp. 567–590, 8 1995.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[13] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[14] S. Pattanayak, *Pro Deep Learning with TensorFlow*. 2017.

[15] H. Wang and B. Raj, "On the origin of deep learning," 2017.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[17] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, 2011.

[18] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.

[19] S. Anwar, W. Sung, and C. Science, "Large batch training of convolutional networks with layer-wise adaptive rate scaling," *Iclr '18*, vol. 1, 2017.

[20] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[21] A. Bhardwaj, W. Di, and J. Wei, *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. 2018.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[23] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. van Essen, A. A. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," vol. 2016-Decem, pp. 770–778, 2016.

[25] A. A. Patel, *Hands-On Unsupervised Learning Using Python.* 2019.

[26] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," 2019.

[27] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," 2020.

[28] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018.

[29] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," vol. 32, 2019.

[30] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," vol. 9907 LNCS, pp. 649–666, 2016.

[31] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," vol. 2019-June, pp. 1920–1929, 2019.

[32] W. Falcon and K. Cho, "A framework for contrastive self-supervised learning and designing a new approach," 2020.

[33] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," 2013.

[34] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," vol. 07-12-June, pp. 815–823, 2015.

[35] O. J. Hénaff, A. Razavi, C. Doersch, S. M. A. Eslami, and A. V. D. Oord, "Data-efficient image recognition with contrastive predictive coding," 2019.

[36] S. J. Pan and Q. Yang, "A survey on transfer learning," 2010.

[37] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," vol. 11141 LNCS, pp. 270–279, 2018.

[38] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?," vol. 2019-June, pp. 2656–2666, 2019.

[39] A. Hocking, J. E. Geach, Y. Sun, and N. Davey, "An automatic taxonomy of galaxy morphology using unsupervised machine learning," *Monthly Notices of the Royal Astronomical Society*, vol. 473, pp. 1108–1129, 2018.

[40] M. Siudek, K. Małek, A. Pollo, T. Krakowski, A. Iovino, M. Scodeggio, T. Moutard, G. Zamorani, L. Guzzo, B. Garilli, B. R. Granett, M. Bolzonella, S. D. L. Torre, U. Abbas, C. Adami, D. Bottini, A. Cappi, O. Cucciati, I. Davidzon, P. Franzetti, A. Fritz, J. Krywult, V. L. Brun, O. L. Fèvre, D. MacCagni, F. Marulli, M. Polletta, L. A. Tasca, R. Tojeiro, D. Vergani, A. Zanichelli, S. Arnouts, J. Bel, E. Branchini, J. Coupon, G. D. Lucia, O. Ilbert, C. P. Haines, L. Moscardini, and T. T. Takeuchi, "The vimos public extragalactic redshift survey (vipers): The complexity of galaxy populations at $0.4 < z < 1.3$ revealed with unsupervised machine-learning algorithms," *Astronomy and Astrophysics*, vol. 617, 2018.

[41] T. Chattopadhyay, D. Fraix-Burnet, and S. Mondal, "Unsupervised classification of galaxies. i. ica feature selection," 2018.

[42] A. Spindler, J. E. Geach, and M. J. Smith, "Astrovader: Astronomical variational deep embedder for unsupervised morphological classification of galaxies and synthetic image generation," 2020.

[43] G. Martin, S. Kaviraj, A. Hocking, S. C. Read, and J. E. Geach, "Galaxy morphological classification in deep-wide surveys via unsupervised machine learning," *Monthly Notices of the Royal Astronomical Society*, vol. 491, pp. 1408–1426, 2020.

[44] N. O. Ralph, R. P. Norris, G. Fang, L. A. Park, T. J. Galvin, M. J. Alger, H. Andernach, C. Lintott, L. Rudnick, S. Shabala, and O. I. Wong, "Radio galaxy zoo: Unsupervised clustering of convolutionally auto-encoded radio-astronomical images," *Publications of the Astronomical Society of the Pacific*, vol. 131, 2019.

[45] C. Ting-Yun, M. Huertas-Company, C. J. Conselice, A. Aragón-Salamanca, B. E. Robertson, and N. Ramachandra, "Beyond the hubble sequence - exploring galaxy morphology with unsupervised machine learning," 2020.

[46] G. Condés-Luna, "Clasificación morfológica de galaxias usando redes neuronales," 2021.

[47] X. P. Zhu, J. M. Dai, C. J. Bian, Y. Chen, S. Chen, and C. Hu, "Galaxy morphology classification with deep convolutional neural networks," *Astrophysics and Space Science*, vol. 364, 4 2019.

[48] A. Mittal, A. Soorya, P. Nagrath, and D. J. Hemanth, "Data augmentation based morphological classification of galaxies using deep convolutional neural network," *Earth Science Informatics*, vol. 13, pp. 601–617, 2020.

[49] I. B. Vavilova, D. V. Dobrycheva, M. Y. Vasylenko, A. A. Elyiv, O. V. Melnyk, and V. Khramtsov, "Machine learning technique for morphological classification of galaxies from the sdss. i. photometry-based approach," *Astronomy and Astrophysics*, 2021.

[50] G. D. Vaucouleurs, "Classification and morphology of external galaxies," 1959.

[51] R. Buta, S. Mitra, G. de Vaucouleurs, and J. C. H. G., "Mean morphological types of bright galaxies," *The Astronomical Journal*, vol. 107, p. 118, 1994.

[52] J. Hedberg, "Galaxies," 2018.

[53] P. B. Nair and R. G. Abraham, "A catalog of detailed visual morphological classifications for 14,034 galaxies in the sloan digital sky survey," *Astrophysical Journal, Supplement Series*, vol. 186, pp. 427–456, 2010.

[54] J. K. Adelman-McCarthy, M. A. Agueros, S. S. Allam, K. S. J. Anderson, S. F. Anderson, J. Annis, N. A. Bahcall, I. K. Baldry, J. C. Barentine, A. Berlind, M. Bernardi, M. R. Blanton, W. N. Boroski, H. J. Brewington, J. Brinchmann, J. Brinkmann, R. J. Brunner, T. Budavari, L. N. Carey, M. A. Carr, F. J. Castander, A. J. Connolly, I. Csabai, P. C. Czarapata, J. J. Dalcanton, M. Doi, F. Dong, D. J. Eisenstein, M. L. Evans, X. Fan, D. P. Finkbeiner, S. D. Friedman, J. A. Frieman, M. Fukugita, B. Gillespie, K. Glazebrook, J. Gray, E. K. Grebel, J. E. Gunn, V. K. Gurbani, E. de Haas, P. B. Hall, F. H. Harris, M. Harvanek, S. L. Hawley, J. Hayes, J. S. Hendry, G. S. Hennessy, R. B. Hindsley, C. M. Hirata, C. J. Hogan, D. W. Hogg, D. J. Holmgren, J. A. Holtzman, S. Ichikawa, Željko Ivezić, S. Jester, D. E. Johnston, A. M. Jorgensen, M. Jurić, S. M. Kent, S. J. Kleinman, G. R. Knapp, A. Y. Kniazev, R. G. Kron, J. Krzesinski, N. Kuropatkin, D. Q. Lamb, H. Lampeitl, B. C. Lee, R. F. Leger, H. Lin, D. C. Long, J. Loveday, R. H. Lupton, B. Margon, D. Martinez-Delgado, R. Mandelbaum, T. Matsubara, P. M. McGehee, T. A. McKay, A. Meiksin, J. A. Munn, R. Nakajima, T. Nash, J. Eric H. Neilsen, H. J. Newberg, P. R. Newman, R. C. Nichol, T. Nicinski, M. Nieto-Santisteban, A. Nitta, W. O'Mullane, S. Okamura, R. Owen, N. Padmanabhan, G. Pauls, J. John Peoples, J. R. Pier, A. C. Pope, D. Pourbaix, T. R. Quinn, G. T. Richards, M. W. Richmond, C. M. Rockosi, D. J. Schlegel, D. P. Schneider, J. Schroeder, R. Scranton, U. Seljak, E. Sheldon, K. Shimasaku, J. A. Smith, V. Smolčić, S. A. Snedden, C. Stoughton, M. A. Strauss, M. SubbaRao, A. S. Szalay, I. Szapudi, P. Szkody, M. Tegmark, A. R. Thakar, D. L. Tucker, A. Uomoto, D. E. V. Berk, J. Vandenberg, M. S. Vogeley, W. Voges, N. P. Vogt, L. M. Walkowicz, D. H. Weinberg, A. A. West, S. D. M. White, Y. Xu, B. Yanny, D. R. Yocum, D. G. York, I. Zehavi, S. Zibetti, and D. B. Zucker, "The fourth data release of the sloan digital sky survey," *The Astrophysical Journal Supplement Series*, vol. 162, pp. 38–48, 2006.

[55] M. Blanton, E. Kazin, G. Zhu, A. Price-Whelan, J. Moustakas, D. Muna, R. Yan, and B. Weaver, "Nasa-sloan atlas."

[56] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," vol. 11218 LNCS, 2018.

[57] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," vol. 2017-January, 2017.

[58] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," *arXiv*, 4 2021.

[59] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," vol. 2017-December, 2017.

[60] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, 2017. https://distill.pub/2017/feature-visualization.