

TRABAJO DE FIN DE GRADO  
GRAO EN ENXEÑERÍA INFORMÁTICA  
MENCIÓN EN COMPUTACIÓN



## **Identificación de Ratas de Laboratorio mediante Visión Artificial y Aprendizaje Máquina**

Estudiante: Diego Antonio López López  
Tutor: Álvaro Rodríguez Tajés  
A Coruña, Septiembre de 2023

*Este proyecto está dedicado a:*

*A mis padres quienes con su esfuerzo, amor y apoyo me permitieron finalizar este trabajo. Gracias por siempre apoyarme en mis momentos más difíciles y recordarme que siempre existe un camino hacia adelante.*

*A mis tías Belén y Ana por su apoyo incondicional, las cuales nunca dudaron que podría realizar este proyecto. A toda mi familia que siempre estuvo ahí para escucharme y comprenderme.*

*Finalmente quiero dedicar este trabajo a mis amigos Luis y Juan por levantar mi ánimo cuando yo solo no podía, y cuando más lo necesitaba.*

## Agradecimientos

*“Quiero agradecer a todas las personas que me han ayudado a la hora de emprender este esfuerzo; desde mis padres que me permitieron concentrarme en este proyecto a tiempo completo y sufragaron los costes del desarrollo y desempeño de los sistemas empleados. También a todos los miembros de mi familia que nunca han dudado de que podría lograrlo y, por supuesto, a mi tutor Álvaro Rodríguez, por dedicar tantas horas ayudándome para que este proyecto tomara la forma correcta.”*

## **Resumen**

En la actualidad, la experimentación animal continúa vigente: en la investigación y monitorización del comportamiento, la farmacología y la ecotoxicología, en los estudios de los efectos del cambio climático, en los estudios de procesos de aprendizaje, así como en la monitorización de granjas. Considerando necesario e importante acompañar la investigación mediante la experimentación animal, es relevante mencionar los programas de seguimiento de imágenes. Mediante la determinación de la posición de los animales, cada imagen, y relacionando las posiciones del animal a lo largo del tiempo para generar su trayectoria, podemos extraer información de gran utilidad. El problema reside en relacionar cada posición con el animal que le corresponde cuando estudiamos múltiples individuos. Los momentos críticos son los instantes de oclusión entre individuos o con el entorno. En este proyecto estudiaré y validaré las técnicas de aprendizaje máquina y transferencia del conocimiento para el estudio de ratas de laboratorio en imágenes. El aprendizaje máquina son un grupo de técnicas capaces de aprender relaciones entre datos. La metodología de la transferencia del conocimiento se basa en la reutilización de técnicas previamente entrenadas en tareas similares. Mediante la velocidad del entrenamiento del aprendizaje por transferencia, en contraposición con enfoques tradicionales de aprendizaje máquina, estas técnicas suponen una optimización sustancial en el tiempo de entrenamiento de los modelos. Así, el resultado fundamental de este trabajo, es demostrar la viabilidad de las técnicas de aprendizaje máquina con transferencia del conocimiento. para identificar animales de laboratorio como las ratas en videos de ensayos animales.

## **Abstract**

Today, animal experimentation continues: in behavioural research and monitoring, pharmacology and ecotoxicology, in studies of the effects of climate change, in studies of learning processes, as well as in farm monitoring. Considering it necessary and important to accompany research through animal experimentation, it is relevant to mention image-tracking programs. By determining the position of the animals in each image and relating the animal's positions over time to generate its trajectory, we can extract useful information. The problem arises when relating each position to the corresponding animal when we study multiple individuals. The critical moments occur in the moments of occlusion between individuals or with the environment. In this research, I will study and validate machine learning and knowledge transfer techniques for the study of laboratory rats in images. Machine learning is a group of techniques capable of learning relationships between data. The knowledge transfer methodology based on the reuse of techniques previously trained in similar tasks. Through the speed of transfer learning training, as opposed to traditional machine learning approaches, these techniques provide substantial optimization in model training time. Thus, the fundamental result of this work is to demonstrate the feasibility of machine learning techniques with knowledge transfer to identify laboratory animals such as rats in animal testing videos.

**Palabras Clave:**

- Identificación Animal
- Ratas de Laboratorio
- Visión Artificial
- Procesamiento de Imágenes
- Aprendizaje Máquina,
- Transferencia del conocimiento.
- Redes Neuronales

**Key Words:**

- Animal Identification,
- Laboratory Rats
- Computer Vision
- Image Processing
- Machine Learning
- Transfer Learning
- Neural Networks



# Índice General

---

<b>INTRODUCCIÓN.....</b>	<b>7</b>
<b>1. ESTADO DEL ARTE.....</b>	<b>9</b>
<b>2. FUNDAMENTOS METODOLÓGICOS.....</b>	<b>12</b>
2.1. VISIÓN ARTIFICIAL Y PROCESADO DE IMÁGENES.....	12
2.1.1. <i>Imágenes</i> .....	12
2.1.2. <i>Pre procesado de imágenes</i> .....	14
2.1.3. <i>Técnicas de seguimiento visual</i> .....	21
2.2. APRENDIZAJE MÁQUINA.....	25
2.2.1. <i>Clasificación</i> .....	25
2.2.2. <i>Arquitecturas de Aprendizaje Máquina</i> .....	25
2.2.3. <i>Redes neuronales</i> .....	26
2.2.4. <i>Aprendizaje Profundo</i> .....	30
2.2.5. <i>Transferencia de conocimiento</i> .....	34
2.3. EVALUACIÓN DE DATOS.....	36
2.3.1. <i>Conjuntos de validación</i> .....	36
2.3.2. <i>Validación cruzada</i> .....	36
2.3.3. <i>Métricas de desempeño</i> .....	37
<b>3. METODOLOGÍA.....</b>	<b>41</b>
3.1. JUSTIFICACIÓN METODOLÓGICA.....	41
3.2. PLANIFICACIÓN ESPIRAL.....	41
3.3. PLANIFICACIÓN.....	42
3.3.1. <i>Ciclo de apertura</i> .....	42
3.3.2. <i>Ciclo 2: Introducción a los datos reales</i> .....	43
3.3.3. <i>Ciclo 3: Prueba de comparación de modelos</i> .....	43
3.3.4. <i>Ciclo de cierre: Análisis del rendimiento de la red</i> .....	44
3.4. EVALUACIÓN DEL DESARROLLO DEL PROYECTO.....	45
<b>4. DESARROLLO.....</b>	<b>46</b>
4.1. ENTORNO SOFTWARE.....	46
4.1.1. <i>Descarga del intérprete</i> .....	46
4.1.2. <i>Librerías</i> .....	46
4.1.3. <i>Otros recursos</i> .....	47
4.2. DESARROLLO DEL PROYECTO.....	48
4.2.1. <i>Pre-proyecto y prueba de concepto con dataset CIFAR-10</i> .....	48
4.2.2. <i>Ensayo de Exploración de Parámetros</i> .....	48
4.2.3. <i>Prueba de comparación de modelos</i> .....	49
4.2.4. <i>Análisis del rendimiento de la red</i> .....	50
4.2.5. <i>Pruebas con imágenes en blanco y negro</i> .....	50
<b>5. RESULTADOS.....</b>	<b>51</b>
5.1. ENSAYO DE EXPLORACIÓN DE PARÁMETROS.....	51
5.1.1. <i>Epochs</i> .....	51
5.1.2. <i>Tamaño de la Imagen</i> .....	51
5.1.3. <i>Tamaño de la muestra</i> .....	52
5.2. PRUEBA DE COMPARACIÓN DE MODELOS.....	54
5.3. ANÁLISIS DEL RENDIMIENTO DE LA RED.....	54
5.4. PRUEBAS CON IMÁGENES EN BLANCO Y NEGRO.....	54
<b>6. CONCLUSIONES.....</b>	<b>56</b>
<b>APÉNDICES.....</b>	<b>57</b>

6.1. RESULTADOS COMPLETOS DE LAS PRUEBAS.....	58
6.1.1. <i>Epochs</i> .....	58
6.1.2. <i>Tamaño de imagen</i> .....	58
6.1.3. <i>Tamaño de muestra</i> .....	61
6.1.4. <i>Prueba cambiando el número de individuos y de clases</i> .....	67
6.1.5. <i>Pruebas con imágenes en blanco y negro</i> .....	68
<b>RELACIÓN DE ACRÓNIMOS.....</b>	<b>70</b>
<b>GLOSARIO.....</b>	<b>71</b>
<b>BIBLIOGRAFÍA.....</b>	<b>75</b>



# Índice de figuras

---

Figura 1 Reconocimiento facial en gorilas.....	10
Figura 2 Reconocimiento del rostro en elefantes.....	10
Figura 3 a) Patrones dorsales en vacas. b) Aletas dorsales en delfines.....	11
Figura 4 Imagen 8x8 binaria blanco y negro.....	12
Figura 5 Imágenes 8x8 binarias blanco y negro y verde y negro.....	13
Figura 6 Ejemplos formatos para la codificación de imágenes.....	13
Figura 7 Imagen original de rata de laboratorio.....	14
Figura 8: Proceso de padding a una imagen.....	15
Figura 9: Padding en rata de laboratorio.....	15
Figura 10: Ejemplo de imágenes con diferentes niveles de entropía.....	15
Figura 11: Entropía en imagen de rata de laboratorio.....	16
Figura 12 Imagen umbralizada.....	16
Figura 13 Imagen de dilatación binaria.....	17
Figura 14 Imagen de erosión binaria.....	17
Figura 15 Imágenes de cerrado binario.....	18
Figura 16 Imagen de resultado del cerrado.....	18
Figura 17 Imagen de filtro de medidas.....	19
Figura 18 Imagen de sustracción de máscaras de una imagen.....	19
Figura 19 Imagen del resultado del proceso de Masking.....	19
Figura 20 Imagen de la rata antes del área de opening.....	20
Figura 21 Imagen de la rata después del área de opening.....	20
Figura 22 Imagen de la rata de laboratorio con el rectángulo envolvente.....	20
Figura 23 Imagen de la rata como resultado del proceso de visión artificial.....	21
Figura 24 Figura de sustracción de fondo.....	22
Figura 25 Imagen de localización de rata.....	22
Figura 26 Imagen de trayectoria generada.....	23
Figura 27 Imagen de Mapa de calor generado en función de las trayectorias del animal en el espacio.....	24
Figura 28 Imagen ejemplo de la arquitectura de un rna.....	27
Figura 29 Patrones de ejemplo.....	27
Figura 30 Ejemplo de generación de predicciones.....	28
Figura 31 Red de neurona.....	28
Figura 32 Funciones sigmoide, tanh y ReLU.....	28
Figura 33 a) Figura de imagen de convolución. b)Figura de imagen como función.....	30

Figura 34 Figura de Operadores de Sobel.....	31
Figura 35 Imagen original.....	31
Figura 36 Imagen de Bordes horizontales.....	31
Figura 37 Imagen Bordes Sobel.....	32
Figura 38 Imagen bordes verticales.....	32
Figura 39 Imagen de número 3 dibujado a mano.....	32
Figura 40 Ejemplo de la pérdida de influencia según la profundización en la red.....	34
Figura 41 Proceso de shortcut o skipping.....	34
Figura 42 K-Fold en 10 conjuntos.....	36
Figura 43 K-Fold con mala partición.....	37
Figura 44 Diagrama de desarrollo en espiral.....	41
Figura 45 Planificación inicial del proyecto vs desarrollo real.....	45
Figura 46 Cascada de dependencias entre librerías y programas.....	47
Figura 47 Imágenes de las ratas de laboratorio en los tres tamaños ensayados.....	49
Figura 48 Ejemplo de valores de pérdida de la red MobileNet para distintos tipos de muestra.....	53
Figura 49. Pérdida y precisión por epoch para imágenes en blanco y negro.....	55
Figura 50 Curva de la función de pérdida para los ciclos empleados.....	58
Figura 51 Pérdida para diferentes tamaños de imagen MobileNet-V2.....	60
Figura 52 Pérdida para diferentes tamaños de imagen DenseNet.....	60
Figura 53 Pérdida para diferentes tamaños de imagen EfficientNet.....	60

# Índice de tablas

---

Tabla 1 Imágenes y convoluciones.....	33
Tabla 2 Valores de las ecuaciones.....	38
Tabla 3 Métricas de desempeño Multiclase.....	39
Tabla 4 Métricas de desempeño generales.....	40
Tabla 5 Matriz de confusión Epochs RESNET- Tabla 6 Positivos u negativos Epochs red RESNET-50.....	51
Tabla 5 Matriz de confusión Epochs RESNET- Tabla 6 Positivos u negativos Epochs red RESNET-50.....	51
Tabla 7 Métricas del caso de Epochs macro-promediada.....	51
Tabla 8 Métricas del caso de tamaño de imagen 32x32 macro-promediadas.....	52
Tabla 9 Métricas por tamaño de la muestra para el modelo MobileNET-V2.....	52
Tabla 10 Pérdida y exactitud en 25 epochs de entrenamiento.....	58
Tabla 11 Pérdida por tamaño de imagen por red.....	59
Tabla 12 Precisión por tamaño de imagen por red.....	59
Tabla 13 Métricas macro-promediadas para tamaño de imagen 64x64.....	61
Tabla 14 Pérdida para diferentes tamaños de imagen MobileNet.....	61
Tabla 15 Métricas macro-promediadas para tamaños de muestra RESNET.....	65
Tabla 16 Métricas macro-promediadas para tamaños de muestra DenseNet.....	65
Tabla 17 Métricas macro-promediadas para tamaños de muestra EfficientNet.....	65
Tabla 18 Tabla de pérdida y exactitud a 16 epochs para la comparación.....	66
Tabla 19 Tiempo de entrenamiento por epoch por red para la comparación.....	66
Tabla 20 Media de valores para la prueba de comparación.....	66
Tabla 21 Métricas macro-promediadas para la prueba de comparación.....	66
Tabla 22 Precisión por clase y número de datos.....	68
Tabla 23 Pérdida y precisión para los resultados de color y blanco y negro.....	69
Tabla 24 Métricas para la prueba de blanco y negro.....	69

# Índice de ecuaciones

---

Ecuación 1 Imagen de resultado de medidas $n=3$ para el píxel (1,1).....	19
Ecuación 2 Filtro de medias ejemplo paso 1.....	19
Ecuación 3 Filtro de medias ejemplo paso 2.....	19
Ecuación 4 Activación del nodo $h_0$ ecuación.....	28
Ecuación 5 Activación del nodo $h_0$ .....	28
Ecuación 6 Activación del nodo $h_0$ con valores.....	28

# Introducción

---

Como ya he mencionado en la introducción, son numerosos los campos de la ciencia en los que se estudian animales y su comportamiento mediante el uso de programas conocidos como algoritmos de seguimiento. éstos son programas dedicados al seguimiento visual a través del procesamiento de vídeo. dichos softwares calculan las posiciones de animales en imagen y relacionan las múltiples posiciones que se dan en el tiempo para así trazar trayectorias. a partir de los datos obtenidos por los programas de seguimiento es posible calcular datos de mayor complejidad como por ejemplo la velocidad de movimiento del animal, el territorio explorado... es así cómo se consigue trazar una monitorización automatizada.

Los algoritmos de seguimiento presentan problemáticas a la hora de identificar sujetos en el mismo espacio, cuando por ejemplo se producen oclusiones debida colisiones y cruces entre individuos o bien cuando uno de los animales se oculta tras un elemento del fondo. En estas circunstancias se producirá la pérdida de los objetivos, de modo que cuando se recupera la detección ya no es posible asegurar qué animal es cual. Debido al problema de la perseveración de la identidad entre sujetos, los algoritmos de seguimiento actuales presentan severas limitaciones temporales o cuantitativas si consideramos el número de animales que pueden ser objeto de estudio con total certeza. Este hecho dificulta en gran medida el uso de estos sistemas o impide su uso teniendo que contar con herramientas alternativas.

El objeto de la presente investigación es la aplicación del aprendizaje máquina y del aprendizaje profundo en el campo de la identificación animal. En concreto se trata sobre el caso de identificación de roedores, mamífero ampliamente sistematizado y empleado en estudio. Usando vídeos de roedores en jaulas, mediante un proceso de visión artificial; creamos las bases de datos de imágenes requeridas como entrada para las redes; reutilizando redes y entradas ya entrenadas por la librería Keras. Mediante la utilización de las redes se elaborarán una serie de pruebas usando métricas como la función de pérdida, la precisión y la métrica F1. Los objetivos implicados en el desarrollo de este estudio son:

Analizar el estado del arte: Realización de una investigación sobre algoritmos de seguimientos, el aprendizaje máquina, el aprendizaje profundo y la visión artificial; mejorando así el conocimiento del estado de las técnicas, de los algoritmos de seguimiento así como sus ventajas e inconvenientes.

Diseño de experimentos: Diseño y realización de un conjunto de pruebas que permitan seleccionar y optimizar la arquitectura y los meta-parámetros de los modelos a usar, y evaluar su desempeño con validez estadística.

Ajuste de diferentes modelos para la tarea y evaluación de modelos: Durante este proyecto entrenaré una serie de redes como las de la familia Resnet (Residual neural Network) o DenseNet (Densely Connected Convolutional Networks) en la tarea de asignar una identidad a las ratas de laboratorio. Del mismo modo someteremos los trabajos bajo una serie de pruebas de evaluación para estimar objetivamente su desempeño.

Análisis de resultados: Interpretación de los resultados obtenidos del proyecto de manera objetiva, identificar sus vulnerabilidades con el objetivo de implementar mejoras a posibles proyectos futuros

# 1. Estado del arte

---

**A** lo largo de este apartado explicaremos la historia de los algoritmos de seguimiento desde el enfoque tradicional hasta la actualidad, terminando en el estado vigente de las técnicas de identificación animal [1], [3].

En los primeros estudios de observación del comportamiento animal las observaciones debían realizarse de forma manual. Los investigadores, por ejemplo, debían observar y anotar manualmente el comportamiento de cada animal. Este tipo de técnicas requiere la presencia de los investigadores durante la totalidad de la duración del experimento. Esto supone una gran inversión de sesgos, así como la posibilidad de aparición de sesgos.

La prevención de los sesgos es una parte fundamental de la investigación. Incluso el investigador más bienintencionado es susceptible de introducirlos de forma involuntaria durante el registro de datos. Por ejemplo, E-bird, un programa de monitorización sobre las poblaciones de aves silvestres [5]. Haciendo uso de la colaboración de voluntarios para registrar manualmente los avistamientos de aves, y aunque de forma no intencional, los observadores expertos son susceptibles de cometer errores al identificar erróneamente el ave [6]. Este es un ejemplo de cómo los sesgos humanos pueden contaminar los resultados de la investigación. Aunque a veces esto resulte inevitable debido a la naturaleza de la investigación, es una buena práctica tratar de reducir el error humano a su más mínima expresión.

Alrededor de 1975 empiezan a realizarse los primeros estudios que hacen uso de grabadoras de vídeo para la automatizar la recogida de datos [9], [11]. Los logros de estas investigaciones se manifestaron en una constante vigilancia de los individuos con la posibilidad de poder revisar posteriormente los datos. Sin embargo, estos sistemas continúan requiriendo de la revisión manual de las grabaciones por parte de investigadores a la hora de generar las anotaciones.

Más adelante, sobre la década de los 80, se incorpora el uso de microprocesadores al campo del seguimiento animal. Por ejemplo, el trabajo de D.B. Dusenbery [12] se usa uno de estos circuitos [13] para detectar a los nematodos mediante el uso de la técnica conocida como thresholding. Este programa, a pesar de haber facilitado en gran medida la investigación, también presentó un gran número de contrariedades como la gran sensibilidad a cambios en la iluminación y precisa de imágenes de alto contraste; lo que no lo hace apto o aplicable para un gran número de escenarios.

Alternativas vigentes [14] y ampliamente empleadas a día de hoy [15] consiste en marcar a los individuos para facilitar su seguimiento. De esta forma fue posible avanzar notablemente en el campo del seguimiento: realizar la identificación requiere marcar a todos y cada uno de los individuos, lo cual conlleva horas de trabajo y puede repercutir en los resultados del estudio.

En la actualidad, existen multitud de sistemas para el seguimiento animal, como por ejemplo en trabajo de N.M Lind [16] donde se utilizan varias técnicas que propician el seguimiento automático como la sustracción de fondo.

En los algoritmos de seguimiento, recientemente se ha analizado el uso de sistemas de inteligencia artificial como la detección, el trazado de trayectoria o el análisis de los datos:

La tarea de detección es la encargada de determinar la posición del animal en la imagen. Puede dividirse en sub tareas como la sustracción de fondo o la segmentación. Existen programas que usan el aprendizaje máquina para realizar esta tarea como por ejemplo *MouseTrack* [17] que proporciona la herramienta de separación de fondo.

El análisis de datos es una labor encargada de extraer información de más alto nivel a partir de la información de seguimiento. Algunos ejemplos son el cálculo del territorio explorado, la velocidad del animal o su orientación. En este cometido, una aplicación de inteligencia artificial a emplear sería, por ejemplo, la detección de las partes del animal; para así estimar su postura.

El trazado de la trayectoria consiste en relacionar las posiciones de cada animal a lo largo del tiempo. En esta tarea los modelos de aprendizaje máquina tienen valor predictivo pues pueden usarse para predecir dónde estará el objeto en el siguiente instante [19], [20]. Otro enfoque en el aprendizaje máquina puede ser útil, en la asignación de trayectorias cuando múltiples individuos colisionan. Mediante el entrenamiento de modelos para el reconocimiento de imágenes como las redes de aprendizaje profundo, basándose en las imágenes es conocida (ABCTracker), (E. Itskovits, 2017) y (Thomas, 2019).

Otros tipos específicos en el enfoque de la identificación es el uso de técnicas basadas en el reconocimiento facial [24], [26] o en el reconocimiento de otras características localizadas en la cabeza del animal [27], [28]. Todos estos estudios se basan en la premisa en la que los individuos pueden reconocerse mediante el rostro. Un inconveniente esencial de estos sistemas son las situaciones donde la cabeza del animal no está presente en la imagen o por oclusiones en el entorno.

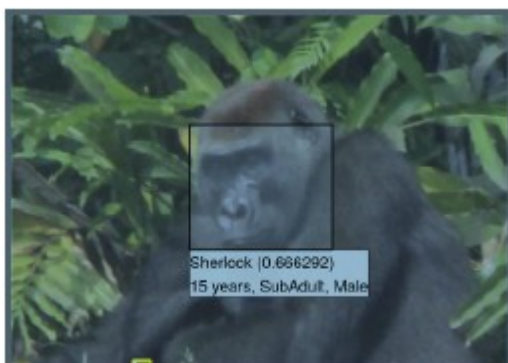


Figura 1 Reconocimiento facial en gorilas

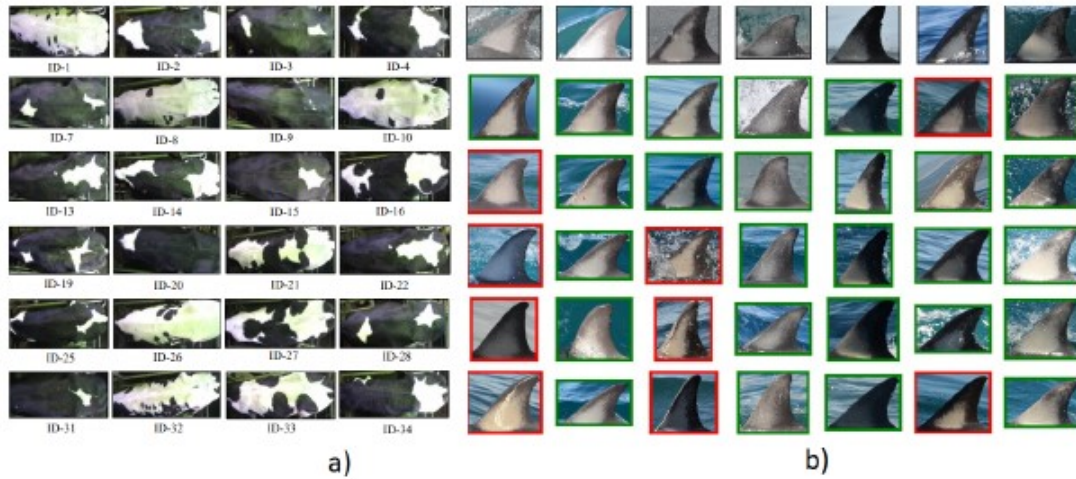


Figura 2 Reconocimiento del rostro en elefantes

Tratando de expandir las técnicas de clasificación de imágenes a escenarios no basados en reconocimiento de rostros, se exploran varios enfoques como el reconocimiento del animal usando imágenes desde todos los ángulos [29].

A pesar de los logros alcanzados en la identificación animal, todavía no se ha hecho una validación formal del uso de modelos de aprendizaje automático para la identificación de nuevos escenarios. En esta investigación utilizaremos las técnicas de aprendizaje máquina y de transferencia del conocimiento para reutilizar redes ya entrenadas, que identifiquen a los individuos en imágenes desde arriba, de un modo semejante a la figura 4. Someteremos nuestros entrenamientos a una serie de pruebas para validar estas técnicas formalmente.





**Figura 3 a) Patrones dorsales en vacas. b) Aletas dorsales en delfines**

A pesar de todos los avances en la identificación animal, todavía no se ha culminado la validación formal de uso de los modelos de aprendizaje automático para la identificación de individuos en escenarios.

En nuestro proyecto utilizaremos técnicas de Aprendizaje Máquina y de Transferencia del conocimiento para reutilizar redes ya entrenadas que identifiquen a los individuos en imágenes de modo similar al que se muestra en las Figuras 3 a [30] y b [31]. Someteremos nuestros entrenamientos a una serie de pruebas para validar estas técnicas formalmente.

## 2. Fundamentos metodológicos

En este apartado se explicarán los conceptos y técnicas clave relacionadas con este proyecto.

### 2.1. Visión artificial y procesamiento de imágenes

La visión artificial es un campo de la Inteligencia Artificial (IA) que permite a los ordenadores y sistemas extraer información significativa de imágenes digitales, vídeos u otras entradas visuales, así como tomar medidas o realizar recomendaciones en función de esa información. Si la IA permite a los ordenadores pensar, la visión artificial les permite ver, observar y comprender (IBM, 2023). Este es un concepto importante porque tanto las técnicas de seguimiento como aquellas que se usan para preparar los datos pertenecen al campo de la visión.

#### 2.1.1. Imágenes

Una imagen digital es la representación bidimensional de una imagen empleando bits, la unidad mínima de información compuesta por dígitos binarios (0,1) AMPLIAR. Existen varios formatos para codificar imágenes. Las imágenes digitales están compuestas por píxeles. Un píxel, los *picture elements* o elementos de imagen son la unidad más elemental de una imagen pudiendo estos comprenderse como los ladrillos que forman una pared o las células que forman un ser vivo.

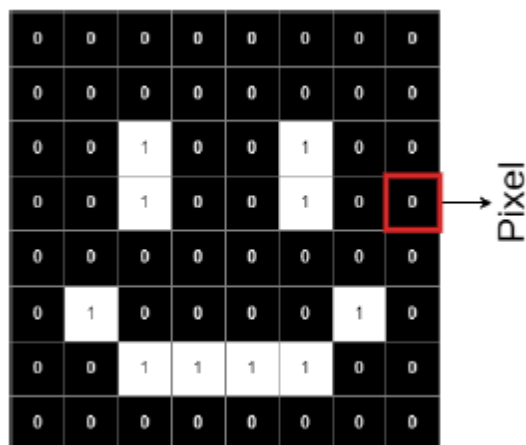


Figura 4 Imagen 8x8 binaria blanco y negro

Pongamos por ejemplo la imagen anterior, Cada cuadrado de esta imagen es un pixel. En este caso es una imagen binaria es decir solo tiene dos tipos de pixel que son 1 y 0.

En este ejemplo se puede observar cómo a partir de una matriz de números se puede formar algo que un ser humano perciba como un dibujo o imagen.

Otro apartado que es importante comprender es que no es el número lo que determina lo que vemos, sino cómo lo interpretamos. Las dos imágenes contienen la misma información, pero si cambiamos la tabla de color, la cual es una tabla que nos dice cómo interpretar los números de la imagen. Obtenemos dos imágenes distintas.

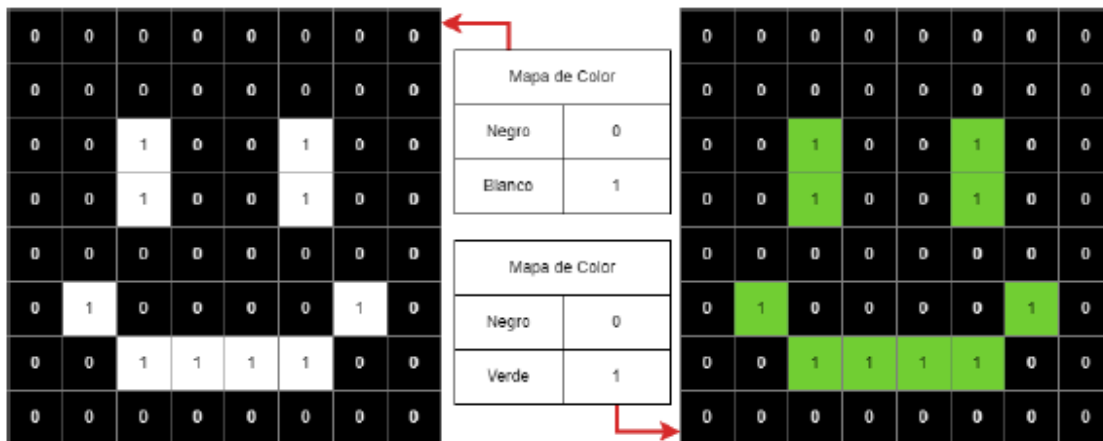


Figura 5 Imágenes 8x8 binarias blanco y negro y verde y negro

### 2.1.1.1. Formatos para la codificación de imágenes

¿Cómo se codifican las imágenes? ¿Cómo se expresa esa percepción visual? Existen múltiples modos de codificar los píxeles en una imagen. A continuación, expongo por orden los más convencionales.



Figura 6 Ejemplos formatos para la codificación de imágenes

- **Monocromo:** Solo se utilizan dos bits donde el 0 representa el negro y el 1 el blanco.
- 4 bits color: Sistema que permite codificar dieciséis colores: 0000, 0001, 0010... No existe un estándar definido sobre qué color debe ocupar cada valor. Este suele determinarse al gusto del usuario pudiendo sufrir múltiples variaciones.
- 8 bits color: En este sistema existen tres canales que representan a razón de tres bits el rojo y el verde y dos bits para el azul. Se elige que canal de azul es el que contará con menos bits asignados puesto que es el color menos sensible al ojo humano, pudiendo así mantener el número de bits por píxel en un múltiplo de dos.
- 8 bits escala de grises: Se trata de imágenes en blanco y negro donde el número representa luminosidad con una gama de doscientos cincuenta y seis tonos entre cero el más oscuro y 256 el más brillante.
- Bits color: Para este formato se asignarán tres canales a rojo, verde y azul de 8 bits cada uno, representando las distintas intensidades de ese color en el píxel. Así se puede obtener una variedad de:  $256 \times 256 \times 256 = 16.777.216$  colores,

### 2.1.1.2. Metadatos

Aparte de la información sobre cómo reconstruir la imagen, estas también pueden tener información adicional, denominada metadatos (datos sobre datos). Dicha información puede variar en cantidad de datos incluida desde imágenes sin información adicional hasta aquellas

con información más pesada que la propia imagen en sí. Algunos campos posibles son el dispositivo que permite capturar la localización, fecha de creación, descripción de una imagen o una etiqueta de clasificación. Descartamos el uso de metadatos en esta investigación, no obstante, es importante considerar su existencia a la hora de realizar trabajos con este tipo de ficheros.

### **2.1.1.3. Vídeo**

En este proyecto se manejan archivos de vídeo, en concreto vídeos de ratas de laboratorio en celdas de condicionamiento durante ensayos de aprendizaje. La fuente ha sido el departamento de psicología del comportamiento de la UNED.

Los vídeos no son más que una sucesión de imágenes, normalmente sucedidas en el tiempo. Este tipo de archivos se someten a estándares que dictan desde cual es el número de fotogramas por segundo que debe tener, hasta el uso de velocidad de transmisión o cómo se codifica. El formato de vídeo utilizado es MPEG4 (Moving Picture Experts Group 4) con expresión “mp4”, obtenidos con cámaras comerciales de la tipología Go Pro. Estos ficheros mediante un proceso de visión artificial se transformarán en una serie de imágenes con las que elaboramos bases de datos para así entrenar las redes.

### **2.1.2. Pre procesamiento de imágenes**

Durante este proyecto se utilizarán múltiples técnicas para el procesamiento de las imágenes y su tratamiento de datos. A continuación, explicaremos como se lleva a cabo el procesamiento de los vídeos de ratas de laboratorio para convertirlos en imágenes aptas para las bases de datos de entrenamiento. Se incluye muestra de imagen original sin proceso de procesamiento alguno.



**Figura 7 Imagen original de rata de laboratorio**

#### **2.1.2.1. Padding**

Como primer paso se expande la imagen para así evitar que se altere el tamaño de la imagen mientras trabajamos con ella. Esto se realiza mediante una transformación llamada padding o relleno, la cual consiste en la expansión de la imagen añadiendo contenido más allá de los bordes; normalmente con ceros o copiando el valor del último pixel de la imagen

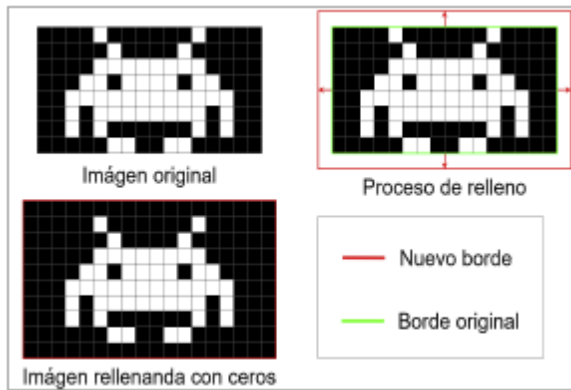


Figura 8: Proceso de padding a una imagen



Figura 9: Padding en rata de laboratorio

### 2.1.2.2. Filtros de entropía local

Para continuar se le aplica a la imagen un filtro de entropía local para aumentar las diferencias entre el fondo y la rata de laboratorio. La entropía local es un concepto relacionado con la complejidad contenida en un conjunto de píxeles vecinos. Por complejidad nos referimos a la diferencia de intensidad de los valores de los píxeles. El filtro de entropía detecta las variaciones sutiles entre estos vecinos y a partir de éstos se calcula la entropía. Así pues, zonas uniformes con poca diferencia en la intensidad entre píxeles vecinos tendrán baja entropía y las zonas con mayores diferencias de intensidad entre píxeles vecinos tendrán mayor entropía.

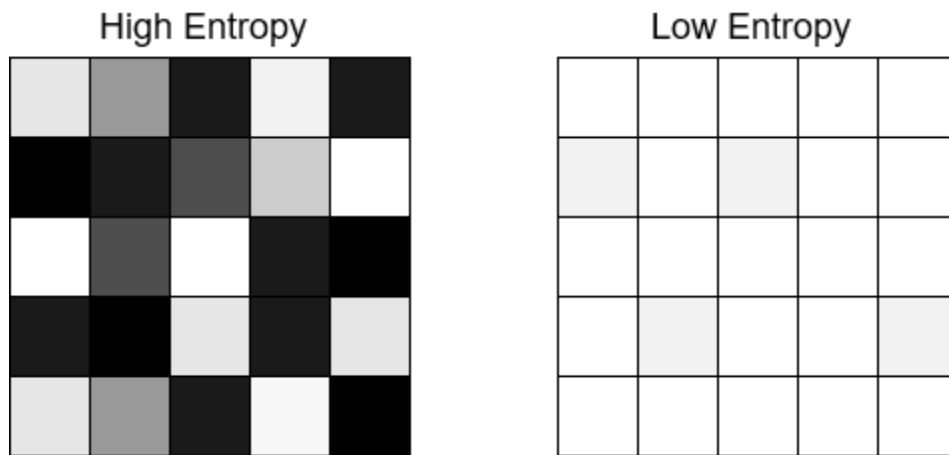


Figura 10: Ejemplo de imágenes con diferentes niveles de entropía



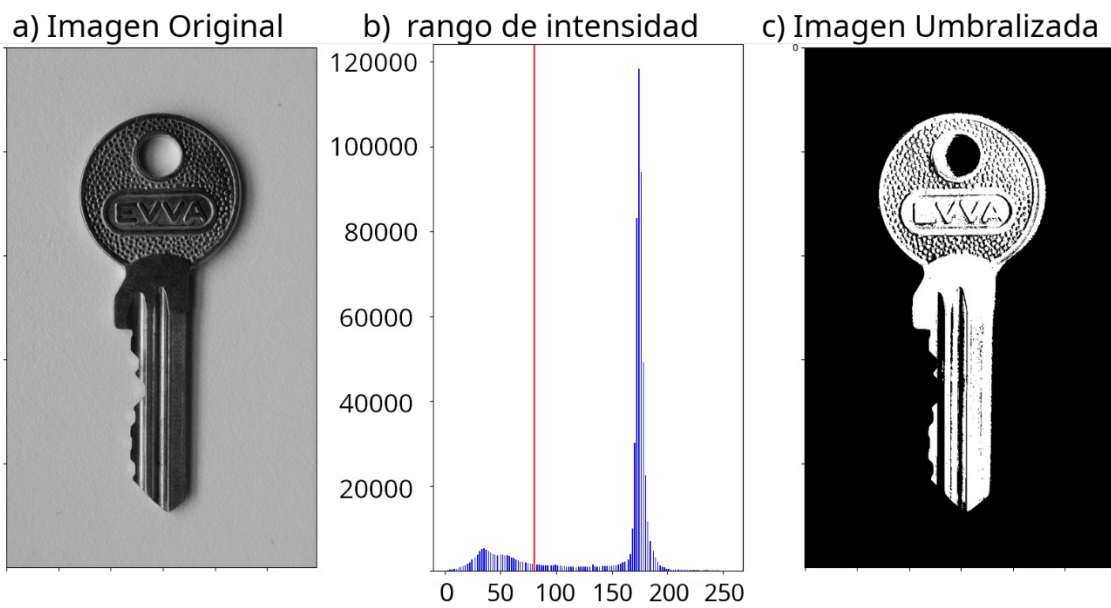
Figura 11: Entropía en imagen de rata de laboratorio

### 2.1.2.3. Normalización

Tras el proceso de filtro de entropía local a continuación se normaliza la imagen. Este es un proceso donde se cambia el rango de intensidad de los píxeles, el cual es el número de valores que abarcan desde el máximo al mínimo. Por ejemplo, pasar de imágenes donde sus valores máximos y mínimos son (25,150) y (75,250) respectivamente y cambiando sus valores pasen todas a ocupar (0,255). Se realiza este proceso para minimizar las diferencias entre imágenes debido a cambios en la iluminación entre imágenes.

### 2.1.2.4. Threshold o umbral

Se realiza después de una operación de threshold o umbral con el objetivo de separar la rata y el fondo. Esta operación consiste en pasar los píxeles a valores binarios (0,1), (true, false), realizando una comparación contra un valor umbral. Si los valores de intensidad del píxel son mayores que el umbral pasaría a valor 1 y si el valor es menor pasaría a valor 0.



### 2.1.2.5. Cerrado

Se realiza una operación morfológica de cerrado sobre la imagen umbralizada para cerrar los huecos. La operación de cerrado es una operación compleja que consiste en dos procesos más simples como la dilatación y la erosión.

### 2.1.2.6. Dilatación

Es una operación morfológica básica. Inicialmente ideada para imágenes binarias y luego expedida a imágenes en escala de gris. En nuestro caso solo usamos la versión binaria. En estas operaciones se combina una imagen con elemento estructurante (matriz), mientras esta se va desplazando por la imagen, dando como resultado una nueva imagen formada y compuesta por todos los puntos donde al menos existe un punto.

Como se puede observar en la siguiente imagen (izquierda) y el siguiente elemento estructurante (centro) dando lugar a su combinación como resultado (derecha).

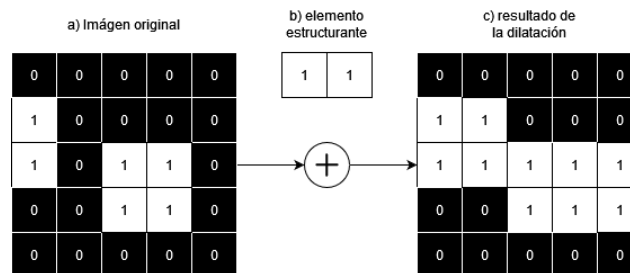


Figura 13 Imagen de dilatación binaria

### 2.1.2.7. Erosión

La erosión es otra de las operaciones morfológicas básicas.

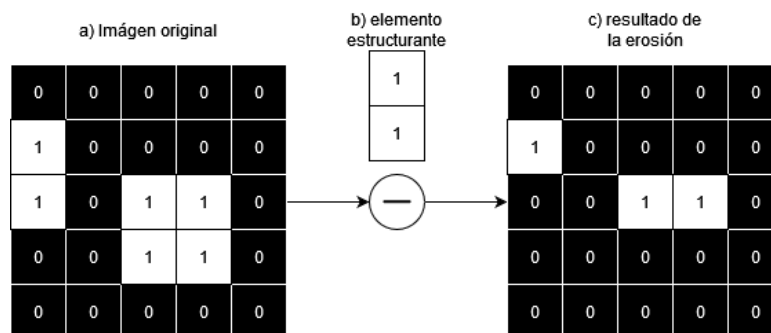


Figura 14 Imagen de erosión binaria

### 2.1.2.8. Cerrado binario

Cuando combinando ambas operaciones realizando primero la dilatación y luego la erosión el resultado es la operación de cerrado binario

Esta operación consta de realizar primero una dilatación para cerrar huecos y a continuación una operación de erosión para devolver la forma a la imagen.

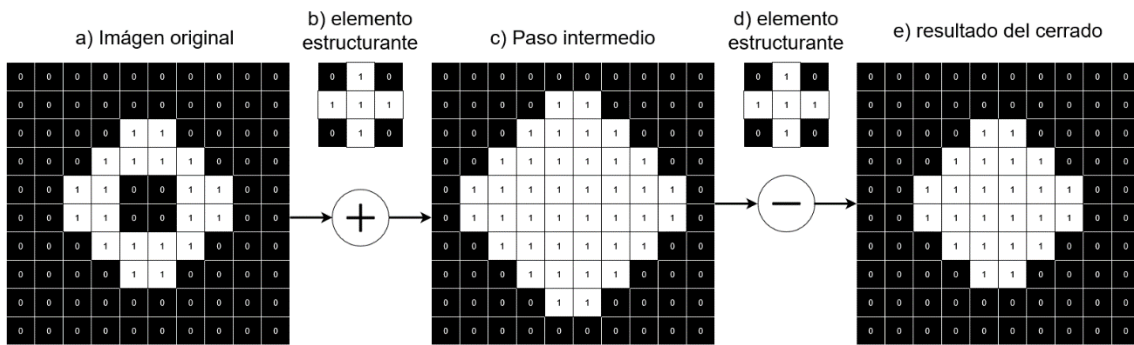


Figura 15 Imágenes de cerrado binario

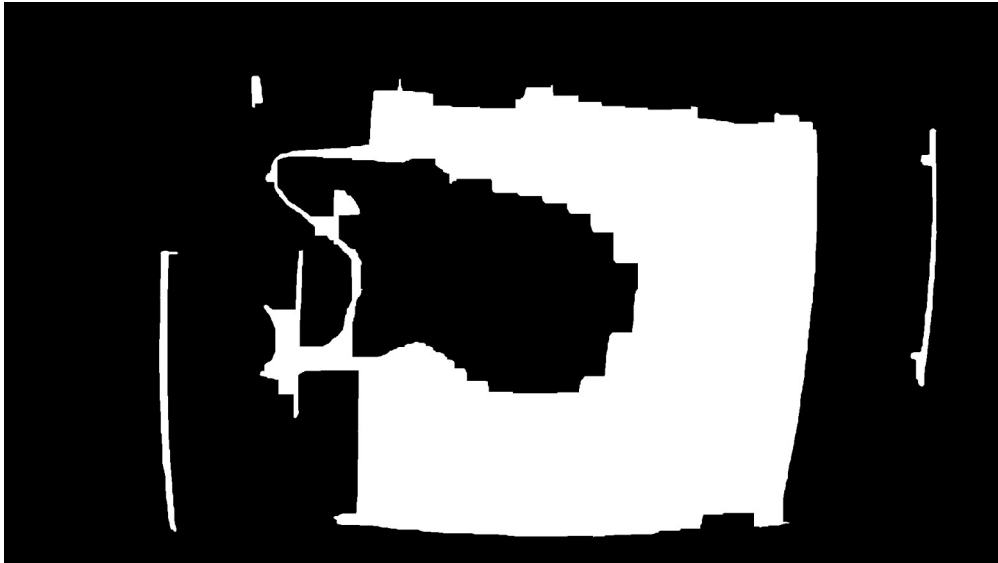


Figura 16 Imagen de resultado del cerrado

### 2.1.2.9. Filtro de medias

Para la aplicación de técnicas de suavizado a través de un filtro de medias, se consigue reducir los problemas de valores atípicos conocidos como ruido. El filtro consiste en formar una nueva imagen con la media aritmética de los píxeles de vecindad  $n$ , más el propio píxel de la imagen original. Por ejemplo, para una imagen como la de la siguiente figura se pasa de la imagen de la izquierda a la de la derecha donde cada píxel representa la media de los valores  $n$  cercanos a la imagen anterior en esa misma posición.





Figura 17 Imagen de filtro de medidas

El resultado del filtro de medias  $n = 3$  para el píxel  $(1,1)$  se calcularía como:

$$x'_{(1,1)} = \frac{\sum_{j=-n, k=-n}^{j=n, k=n} x_{(j,k)}}{n}$$

Ecuación 1 Imagen de resultado de medidas  $n=3$  para el píxel  $(1,1)$

$$x'_{(1,1)} = \frac{x_{(0,0)} + x_{(0,1)} + x_{(0,2)} + x_{(1,0)} + x_{(1,1)} + x_{(1,2)} + x_{(2,0)} + x_{(2,1)} + x_{(2,2)}}{9}$$

Ecuación 2 Filtro de medias ejemplo paso 1

$$x'_{(1,1)} = \frac{0.5 + 0.5 + 0.5 + 0.5 + 1.0 + 1.0 + 0.5 + 0.2 + 0.2}{9} = 0.54$$

Ecuación 3 Filtro de medidas ejemplo paso 2

Como resulta del ejemplo esta es una de las operaciones que necesita el uso de padding para mantener el tamaño de la imagen.

### 2.1.2.10. Sustracción de máscaras

Así mismo, disponemos de unas máscaras constituidas imágenes binarias que son utilizadas para sustraer la parte de la imagen perteneciente a la imagen de la rata de laboratorio. Este es un simple proceso de resta donde ambas imágenes son binarias y simplemente se sustrae la máscara a la imagen original.

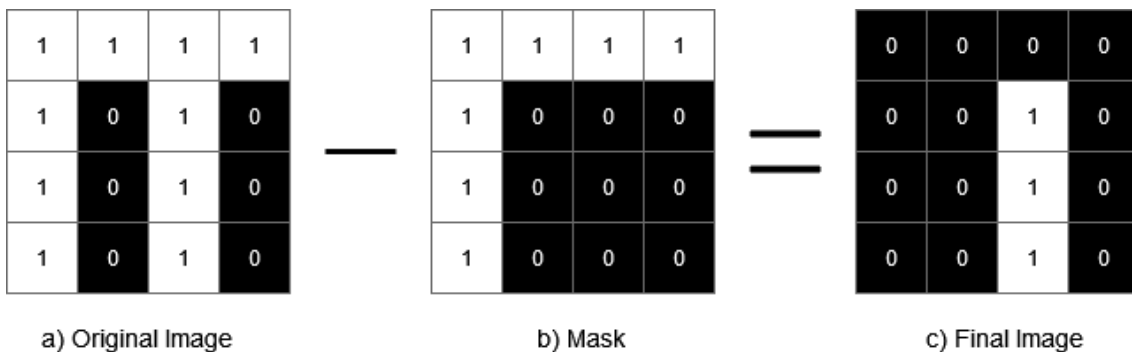


Figura 18 Imagen de sustracción de máscaras de una imagen



Figura 19 Imagen del resultado del proceso de Masking

### 2.1.2.11. Apertura en Área

Una vez realizada la sustracción de la máscara se le aplica a la imagen una operación de apertura en área. Este proceso ofrece resultados similares a la apertura, pero en lugar de

depender de un elemento estructurante, de este modo, se eliminan los objetos cuya área sea menor a un umbral. Este proceso se realiza para eliminar los objetos pequeños de la imagen (al no ser relevantes ya que buscamos a la rata siendo esta un elemento grande). Esto ayuda a reducir el número de operaciones en el siguiente paso (borrando una gran cantidad de objetos que claramente no son el roedor, agilizando ampliamente el proceso de threshold del por área).

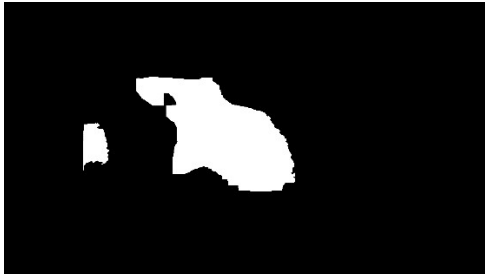


Figura 20 Imagen de la rata antes del área de opening



Figura 21 Imagen de la rata después del área de opening

Como puede visualizarse en la imagen solo los objetos de un área superior al umbral quedan en la imagen final.

#### 2.1.2.12. *Umbral de Área detectada*

Una vez reducido el número de áreas a revisar con el área opening, se realiza un proceso de área threshold lo que nos permitirá quedarnos solo con el objeto más grande (la rata de laboratorio).

#### 2.1.2.13. *Rectángulo envolvente*

A continuación, se calcula un rectángulo que encapsule a la rata de laboratorio. Esta área será la utilizar en el siguiente paso.



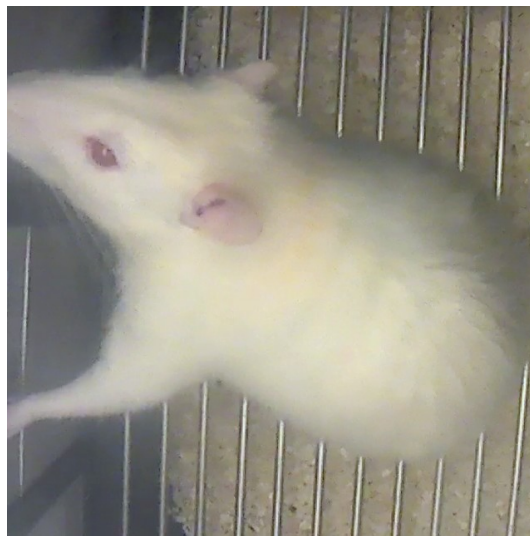
Figura 22 Imagen de la rata de laboratorio con el rectángulo envolvente

#### **2.1.2.14. Muestreo**

A partir del rectángulo envolvente se calcula una matriz tomando las medidas del interior.

#### **2.1.2.15. Cambio de Resolución**

Por último, se realiza una operación que cambia el tamaño de imagen a 640 x 640 píxeles. Este proceso es bastante simple una vez calculada una matriz cuadrada con resolución. Solo se trata esta matriz como una imagen y mediante el método de resolución de openCV con INTER\_AREA, el cual consiste en un nuevo muestreo que utiliza una relación del área para generar cada pixel. En esta nueva imagen píxel se representan los píxeles que existían en la imagen anterior porque pasamos a una imagen más grande a una más pequeña, con lo cual múltiples píxeles conforman cada nuevo píxel. Como solución al problema se genera un nuevo píxel calculando su contribución basándose en el área ocupada por los nuevos resultados.



**Figura 23** Imagen de la rata como resultado del proceso de visión artificial

Estos procesos nos dejan con una imagen de 640x640 píxeles centrada en la rata de laboratorio. Este tipo de imágenes serán las que usemos como base para crear las múltiples bases de datos para entrenar las redes neuronales.

### **2.1.3. Técnicas de seguimiento visual**

Estas técnicas forman parte de la visión artificial ya que extraen información de las trayectorias de objetos en base a imágenes. En el siguiente apartado hablaremos de los algoritmos de seguimiento. Estos son programas que a través de diferentes procesos son capaces de generar las trayectorias de objetos basándose en sucesiones de imágenes.

#### **2.1.3.1. Funcionamiento de los algoritmos de seguimiento**

Para entender qué es lo que se realiza en esta investigación se explicarán los algoritmos de seguimiento [34]. El procedimiento de estos algoritmos se puede dividir en subtarefas. Aunque no todas estas son necesarias y ciertos algoritmos optan por omitir algunas por completo, las tareas habituales que deben realizarse cuando se trata de seguimiento de imágenes son: Adquisición de imágenes, calibración, detección, generación de trayectorias y análisis de datos.

#### **2.1.3.2. Adquisición de imágenes**

En este apartado se determina cómo se obtienen las imágenes del video. Existen dos maneras de abordar el problema, se capturan las imágenes primero para después pasarlas al algoritmo de seguimiento o se capturan en vivo al mismo tiempo que se realiza el seguimiento. ¿Por qué no se capturan en vivo todas las imágenes? La transmisión de video en vivo requiere algoritmos muy rápidos debido a la necesidad de trabajar en muy poco tiempo, lo que conlleva la ventaja de monitoreo en tiempo real y menor uso de almacenamiento (ya que no necesitan mantener los videos originales). Sin embargo, al limitar el tiempo se imposibilita el uso de algoritmos más lentos y complejos. En situaciones complejas dado que todavía existe la necesidad de realizar un seguimiento se podría optar por tener las grabaciones primero para dejar que los algoritmos más complejos las procesen después.

### **2.1.3.3. Calibración**

Este paso sirve de ayuda al programa a la hora de establecer la relación entre los píxeles y la unidad del mundo real. Mientras que algunos programas usan una técnica de calibración conocida como modelo de cámara de pinole [35] que usa un conjunto de patrones planos para resolver las ecuaciones que describen la proyección de un punto en el mundo real al plano de la imagen a través de la lente de una cámara ideal.

La mayoría de las técnicas de calibración son mucho más sencillas. Simplemente se hace uso de una transformación de escala que multiplica las coordenadas de la imagen por una constante. la mayoría de los algoritmos utilizan.

### **2.1.3.4. Detección**

En este paso el algoritmo determina la posición espacial de los animales. Podemos identificar este proceso en dos modalidades diferentes: la sustracción de fondo y la segmentación.

#### **Sustracción de fondo**

En primer lugar, debe eliminarse cualquier característica ambiental que pueda interferir con la detección de animales, habiendo diversos modos de lograr este objetivo.

Una de las aproximaciones más comunes es la técnica de comparar todas las imágenes de vídeo con el animal y una imagen de referencia de fondo. Esta tarea también puede realizarse estimando el fondo mediante un promedio de fotogramas de vídeo. Los sistemas más complejos pueden usar modelos de promedio en desvaneciente, que es una técnica que va graduando una media del fondo de las imágenes para tener en cuenta espacios que cambian con el tiempo.

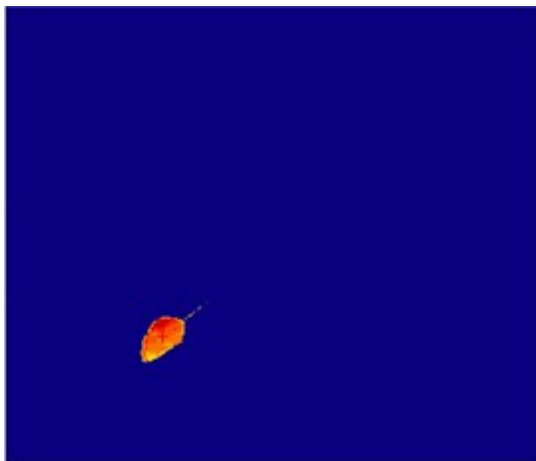


Figura 24 Figura de sustracción de fondo

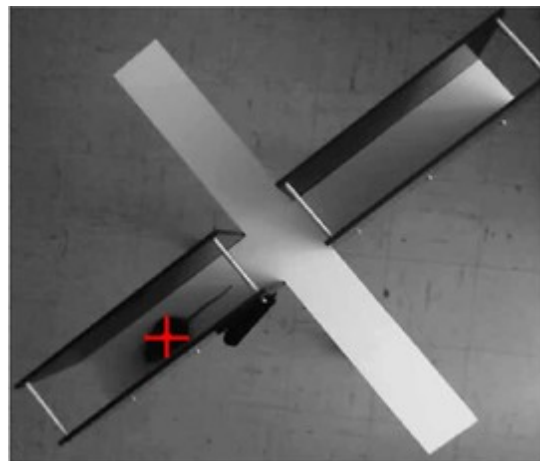


Figura 25 Imagen de localización de rata

## ***Segmentación***

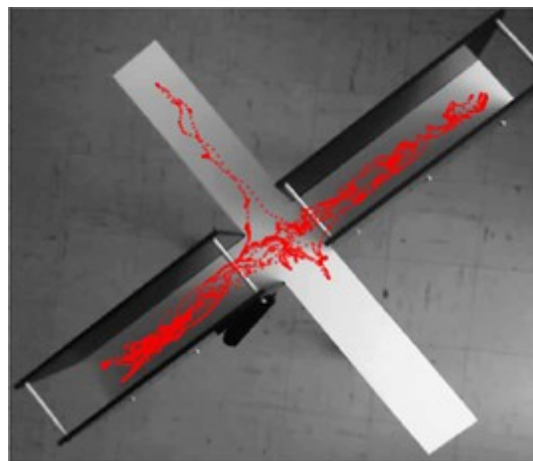
Es un proceso que se realiza inmediatamente después de la sustracción del fondo y suele consistir en una técnica destinada a separar objetos potenciales en la imagen. La técnica de segmentación más utilizada es el Threshold; este método utiliza un valor de referencia para separar las regiones de la imagen con un brillo diferente. Esto es muy eficiente pero también es muy sensible a detectar objetos donde no los hay y a perder objetos que sí estaban presentes.

### ***2.1.3.5. Generación de trayectorias***

Esta es la tarea encargada de relacionar posiciones a lo largo del tiempo. La técnica más común cuando se trata de resolver este problema es el filtro de Kalman, una técnica de corrección de predicción. Esta técnica se basa en predecir la posición del objeto entre fotogramas. Estas estimaciones se basan en asumir que el objeto lleva la misma velocidad y aceleración para poder relacionar las posiciones que pertenecen a una misma trayectoria. La mayoría de los algoritmos emplean alguna versión de esta técnica.

Sin embargo, esta técnica tiene problemas relacionados con el seguimiento de múltiples objetos, ya que existe la posibilidad de que dos objetos puedan chocar y obstruirse entre ellos. Cuando ocurre este escenario, el algoritmo perderá el rastro de los objetos, el algoritmo intentará hallar la identidad comparando la velocidad y la dirección de los animales con valores anteriores. Debido a la probabilidad de que los dos objetos cambien de rumbo después de la colisión, puede asignar mal las identidades.

Una de las soluciones a este problema puede ser marcar a los individuos. Sin embargo, muchas técnicas modernas intentan evitar colocar marcadores o sensores en los animales porque los marcadores pueden ser poco prácticos y, a veces, interferir con el experimento al afectar al comportamiento de los animales [35].



**Figura 26 Imagen de trayectoria generada**

Otros programas utilizan una estrategia basada en un análisis de características de los individuos para analizar la identidad de los animales entre las oclusiones. El análisis no es capaz de rastrear muchos animales o de hacerlo durante mucho tiempo, pero el enfoque es muy útil en experimentos cortos con pequeños grupos de animales. Los desarrolladores recomiendan realizar los experimentos limitándose a cinco individuos y por no más de veinte minutos (Rodríguez A., 2018).

### ***2.1.3.6. Análisis de datos***

El paso de análisis consiste en extraer información de la trayectoria. Esto puede variar desde estadísticas simples como movimiento, velocidad y orientación hasta tareas más complejas como territorio explorado o tasa de actividad. Como por ejemplo en la siguiente imagen donde tenemos un mapa con la actividad del ratón en cada área.

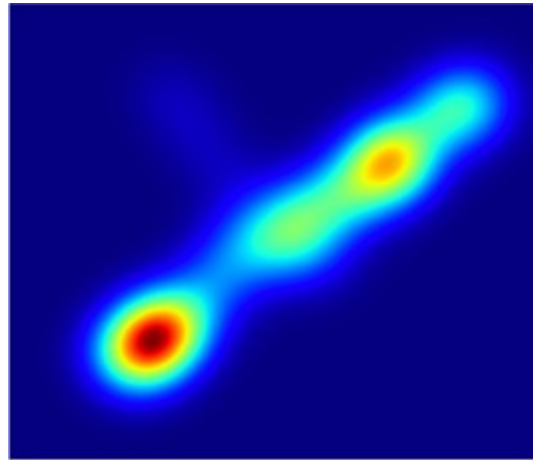


Figura 27 Imagen de Mapa de calor<sup>1</sup> generado en función de las trayectorias del animal en el espacio

### **2.1.3.7. Aprendizaje Máquina y seguimiento**

¿Qué aporta el aprendizaje máquina al seguimiento? Estas técnicas se han utilizado ampliamente en otros campos para automatizar y mejorar tareas que requieren algún tipo de comprensión. En el campo del seguimiento, se pueden emplear por ejemplo en las siguientes tareas:

Separación de fondo – primer plano: Las técnicas de aprendizaje automático se pueden usar para aprender las características del objeto en movimiento que queremos rastrear. De este modo puede separarse del fondo [17].

Generación de trayectorias: En la tarea de generación de trayectorias se podrías usar los modelos de decisión de Markov para pronosticar las posibles trayectorias de los objetos [19].

Análisis de datos: Este enfoque es de utilidad cuando se trata de analizar el contenido, por ejemplo, en este estudio “Estimación rápida de poses de animales utilizando redes neuronales profundas” [18] donde los investigadores utilizan las técnicas de aprendizaje automático para estimar la pose en animales.

Identificación visual de objetos: Estos sistemas también pueden ser entrenados en el reconocimiento de ciertos objetos o partes de un objeto en base a sus características y de esa manera ayudar al rastreo resolviendo conflictos y así ayudando a mantener la identidad del individuo [26], [30], [32]siendo éste el enfoque de la investigación.

---

<sup>1</sup> Mapa de distribución de una variable.

## **2.2. Aprendizaje Máquina**

El aprendizaje máquina son una serie de técnicas pertenecientes a la inteligencia artificial capaces de aprender relaciones entre datos. En esta sección explicaremos los conceptos de este campo relacionados con nuestro proyecto como son las tareas de clasificación, aprendizaje supervisado, aprendizaje por transferencia, redes neuronales y aprendizaje profundo.

Para entender lo que realizamos primero explicamos qué es lo que tratamos de desarrollar en este proyecto. Es decir que problema tratamos de abarcar y qué técnicas utilizaremos.

### **2.2.1. Clasificación**

Lo primero que hay que comprender es lo que intentamos resolver mediante el uso de estas técnicas. En el proyecto se trata de asignar la identidad a dieciséis roedores mediante imágenes. Entonces tratamos con una tarea de clasificación multiclase de imágenes [36].

La clasificación es un tipo de tareas donde el objetivo principal es asignar un identificador al conjunto de datos (como el objetivo de rata de nuestro conjunto de imágenes). Dependiendo del número de clases que se quieran clasificar este se puede dividir en dos categorías clasificación binaria y clasificación multiclase. También se dividen este tipo de problemas dependiendo si el individuo puede o no pertenecer a más de una clase o del número de individuos por clase. Se realizan estas distinciones por la diferencia de técnicas y enfoques empleados a la hora de resolver el problema.

#### **2.2.1.1. Clasificación binaria**

Se considera que un problema es de clasificación binaria si en el conjunto que se quiere segregar solo existen dos clases distintas y el individuo tiene que pertenecer a una de las clases. Como por ejemplo el primer problema que tratamos en el proyecto durante la fase de introducción que se centraba en clasificar imágenes de animales en perros o gatos.

#### **2.2.1.2. Clasificación multiclase**

Por el contrario, si existen tres o más clases se considera que el problema pertenece a la clasificación multiclase. Algunos ejemplos como los que trabajamos en el proyecto son la base de CIFAR 10 donde se identifican 10 tipos de imágenes: aviones, automóviles, pájaros, gatos, ciervos, perros, ranas, caballos, embarcaciones y camiones.

### **2.2.2. Arquitecturas de Aprendizaje Máquina**

En el aprendizaje máquina existen tres tipos principales de enfoques conocidos como aprendizaje no supervisado, aprendizaje supervisado y aprendizaje por refuerzo. Estos se diferencian por cómo afrontan el proceso de entrenamiento.

#### **2.2.2.1. Aprendizaje no supervisado**

Este se llama aprendizaje no supervisado porque usa grupos de datos sin identificación. Estos algoritmos descubren patrones y los agrupan sin necesidad de intervención humana. Esta característica los hace particularmente adecuados para problemas donde se desconoce la solución o esta no es de fácil codificación [37].

#### **2.2.2.2. Aprendizaje supervisado**

Por el contrario, el aprendizaje supervisado se caracteriza por analizar y usar grupos de datos con identificación puesta por los desarrolladores. Este conjunto de datos de entrenamiento incluye entradas y salidas correctas, que permiten que el modelo aprenda con el tiempo. En método es preferido para realizar clasificaciones o predicciones [38].

### **2.2.2.3. *Aprendizaje por refuerzo***

Cambiando completamente el enfoque, el aprendizaje por refuerzo es un paradigma que "imita" la forma de aprender de los seres humanos. En este no se tiene un conjunto de datos con los resultados correctos o ground truth en vez se tienen una serie de casillas que contienen "estado actual" - "acción" - recompensa. Partiendo de estas y mediante interacción con el entorno el agente se adapta y obtiene una política de comportamiento que mejora los resultados [39].

Dadas las condiciones de nuestro proyecto donde disponemos de de una gran cantidad de datos etiquetados en un entorno no interactivo se opta por emplear técnicas de aprendizaje supervisado en concreto nos centraremos en el campo de las redes neuronales.

### **2.2.3. Redes neuronales**

Queremos resolver un problema de clasificación multiclase para lo cual decidimos usar técnicas de aprendizaje supervisado, en concreto dentro de estas técnicas usaremos redes neuronales.

Las redes neuronales esencialmente son modelos inspirados en arquitectura de las redes de neuronas existentes en la naturaleza es decir los sistemas nerviosos de los animales.

#### **2.2.3.1. *Neuronas en la naturaleza***

En la naturaleza los sistemas nerviosos están formados por neuronas. La neurona es la unidad fundamental del sistema nervioso, siendo ésta una célula especializada en recibir y propagar impulsos electroquímicos. Dentro de la neurona la información se transmite con señales eléctricas y entre neuronas con neurotransmisores químicos.

#### **2.2.3.2. *Sistemas nerviosos y comportamientos complejos***

Los sistemas nerviosos se basan en que a partir de la interacción de elementos simples se pueden dar comportamientos más complejos. En ese sentido es parecido, una neurona por sí sola está limitada a transmitir impulsos, pero si estas se organizan de manera adecuada esto da lugar a comportamientos como moverse, respirar, percibir el mundo...

Las redes neuronales buscan replicar esa capacidad que tienen las neuronas de que partiendo de un concepto sumamente simple (recibir un impulso y transmitirlo a otro lugar) se logran realizar tareas más complejas. Así pues, las redes neuronales basan su arquitectura en estas, cambiando las dendritas y los axones por conexiones con un peso que simula la intensidad con que se conectan las neuronas y el soma por un nodo que "contienen" un valor.

#### **2.2.3.3. *Funcionamiento de una red neuronal***

Una red neuronal no es más una que una operación matemática formada por un conjunto de nodos y conexiones entre ellos. Para explicar su funcionamiento usaremos como ejemplo los MLP (multilayer perceptron) que son redes neuronales multicapa "feedforward" es decir que las neuronas están conectadas en un solo sentido de entrada a salida lo que simplifica su entrenamiento. Las redes neuronales se basan en tres componentes fundamentales:



- Capa de entrada: Neuronas que no computan nada pero que almacenan los datos que pasan a capas posteriores (entrada de la red).
- Capas ocultas: Capas que se encargan del proceso de transformación del input a outputs (procesamiento de la tarea).
- Capa de salida: Neuronas que representan la salida del rna (resultado).

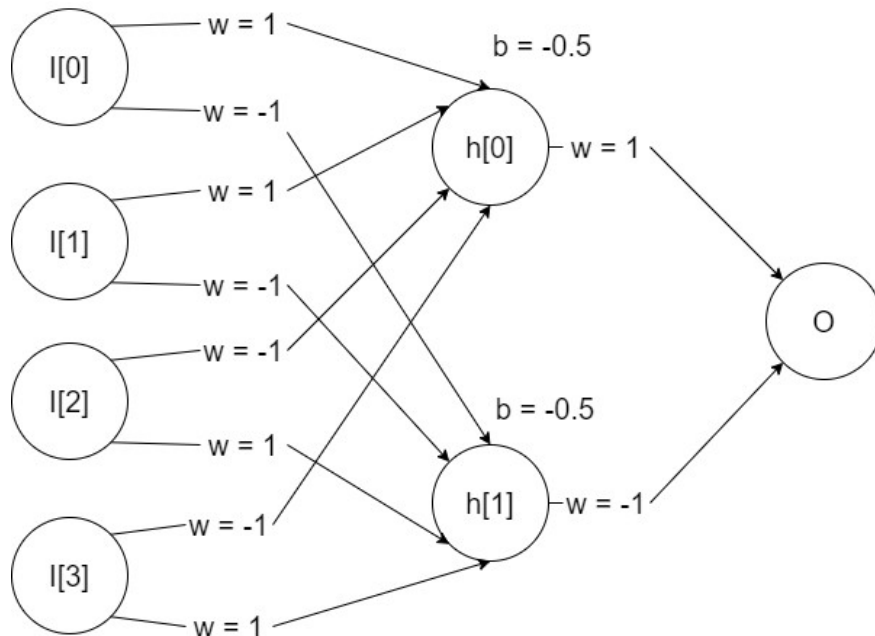


Figura 28 Imagen ejemplo de la arquitectura de un rna

### Generación de predicciones

Los resultados se generan en el momento en el que queremos identificar los dos patrones de la figura 30, donde un valor de 0 representa al primer patrón y 1 representa al segundo.

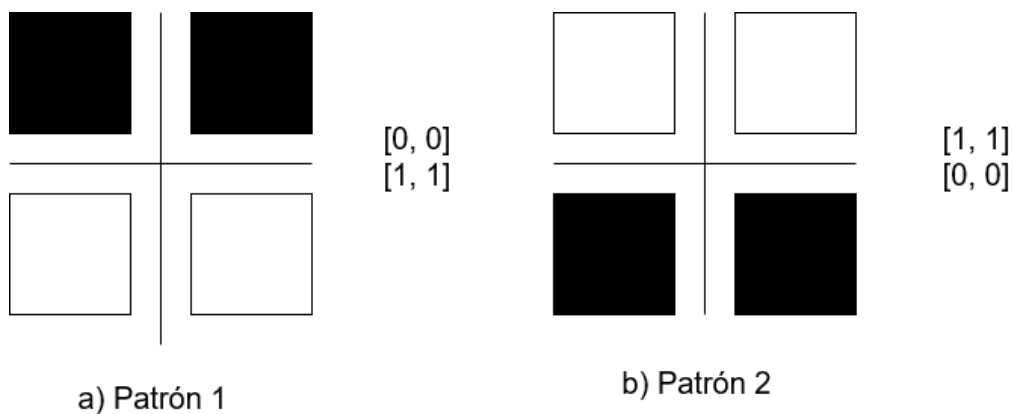
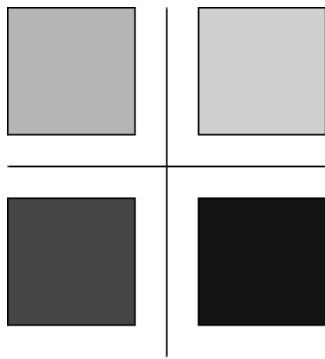


Figura 29 Patrones de ejemplo



Ejemplo

[0.75, 0.80]  
[0.25, 0.10]

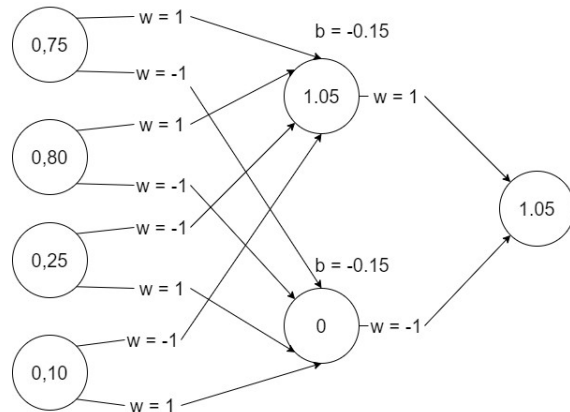


Figura 31 Red de neurona

En el primer paso, el de red o capa de entrada, la imagen entra a la red de neuronas y los nodos de entrada toman como valor la intensidad de la imagen de éstas. Las cuales transmiten el valor a la siguiente capa multiplicando su valor por el peso de la conexión. El resultado para las neuronas de la siguiente capa es el agregado de todas las neuronas que le conectan, más el bias de activación, el cual se impone para evitar que las neuronas se “disparen” con entradas de poca intensidad.

$$ej: h_0 = I_0 * W_{0,0} + I_1 * W_{1,0} + I_2 * W_{2,0} + I_3 * W_{3,0} + b$$

Ecuación 4 Activación del nodo h0 ecuación

$$h_0 = 0.75 * 1 + 0.80 * 1 + 0.25 * -1 + 0.1 * -1 - 0.15 = 1.05$$

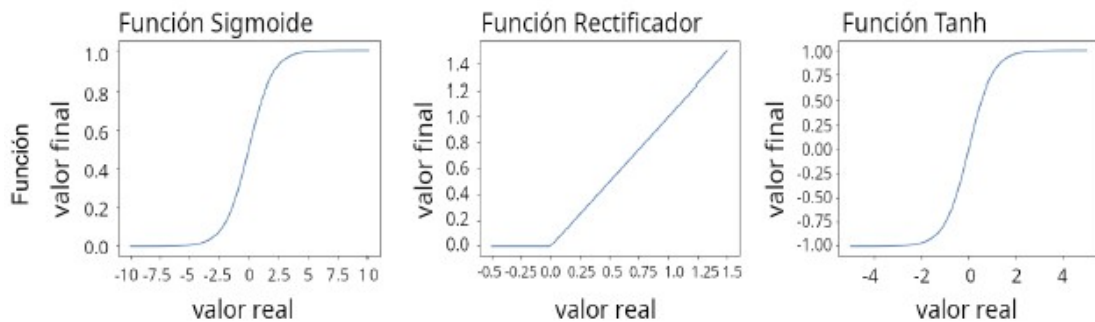
Ecuación 5 Activación del nodo h1

$$h_1 = 0.75 * -1 + 0.80 * -1 + 0.25 * 1 + 0.1 * 1 - 0.15 = -1.35$$

Ecuación 6 Activación del nodo h0 con valores

Otra técnica que evita que se disparen las neuronas es el uso de funciones como Sigmoide, Tanh o ReLU (rectificador).

Figura 32 Funciones sigmoide, tanh y ReLU



Ecuación

$$S = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \quad ReLU = \max(0, x) \quad tanh = \frac{\sinh}{\cosh} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Si continuamos el cálculo hasta llegar a la capa de salida quedaría un resultado igual al mostrado en la figura:

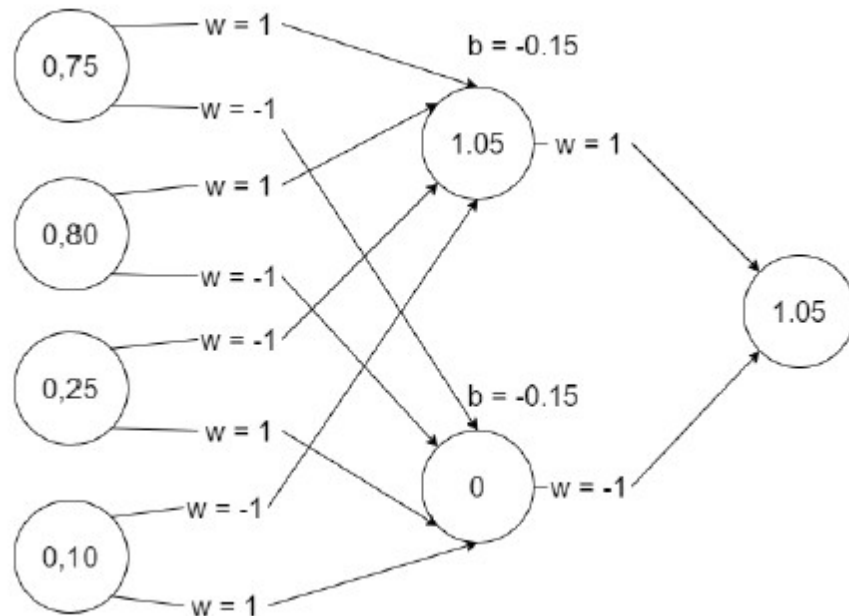


Figura 34. Red con pesos calculados

### 2.2.4. Aprendizaje Profundo

Ahondando algo más en el campo del aprendizaje máquina hablaremos sobre el Aprendizaje Profundo, ya que es a este subcampo al que pertenecen todas las redes empleadas en este proyecto.

El Aprendizaje Profundo es un subconjunto del aprendizaje máquina que se diferencia del aprendizaje clásico por los datos que emplea y los modelos con los que aprende (IMB, 2023).

Los algoritmos de Aprendizaje Profundo normalmente se tratan con conjuntos de datos que no suelen tener estructura y que cuentan con poco pre-procesado. De este modo se elimina la dependencia de expertos humanos. Estos algoritmos pueden determinar por sí mismos cuál es la característica más importante a detectar. Esto en Aprendizaje Máquina debe ser determinado por un experto.

Según el tipo de problema encontramos una variedad de redes neuronales recurrentes, empleadas en problemas secuenciales o series temporales como puede ser el reconocimiento de voz:

CNNs; Convolutional neural networks o redes neuronales de convolución que se utilizan en problemas de visión por computador, detección de patrones o reconocimiento de objetos.

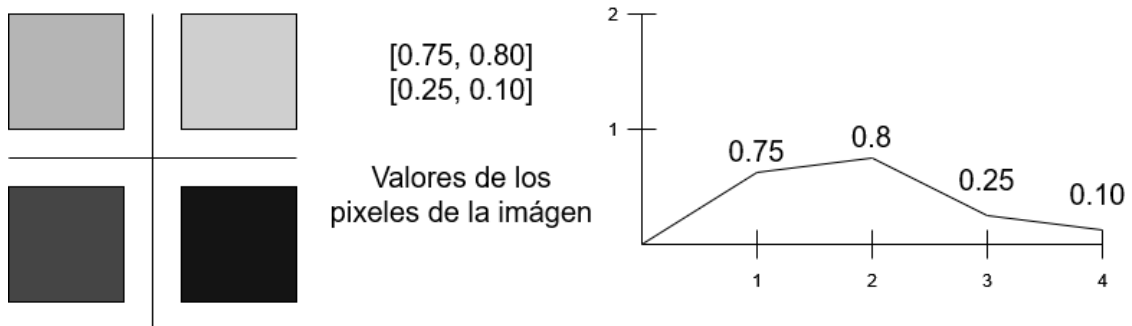
RNNs: Recurrent neural networks o redes neuronales recurrentes, empleadas en problemas secuenciales o series temporales como puede ser el reconocimiento de voz.

#### 2.2.4.1. Redes neuronales de convolución

Al tratar con imágenes este es el tipo de redes que emplearemos en el proyecto. Pero ¿En qué se diferencian estas de las redes convencionales? La diferencia elemental es la existencia de unas capas previas a la red que llevan a cabo operaciones de convolución ¿Qué es una operación de convolución? La convolución es una operación matemática que transforma dos funciones  $f$  y  $g$ , a una tercera función que representa lo que se superponen  $f$  y una versión invertida de  $g$ .

#### 2.2.4.2. Convolución de imágenes

Aplicando este concepto, si interpretamos la imagen como una función donde cada pixel es uno de los valores de la misma, entonces se le puede aplicar un proceso de convolución con los valores de otra función (kernel) para obtener una tercera imagen.



a) Imagen

b) Imagen como función

Figura 33 a) Figura de imagen de convolución. b) Figura de imagen como función

Los kernels tienen una estructura que al convolucionar con la imagen original resta los valores de los píxeles. De este modo si todos los valores de la imagen que se superponen con el kernel son iguales (no estando cerca de un borde), la suma es cero. Pero, si existe un cambio de intensidad (borde) los valores serán positivos si cambiamos de una zona de baja a alta intensidad a alta intensidad, y serán negativos si pasamos de una zona de alta intensidad a otra de baja intensidad.

En esta serie de imágenes se muestra el resultado de la convolución de los kernels anteriores a una imagen. En ellas puede observarse que sólo quedan valores en los bordes de los objetos (lugar donde cambia la intensidad de la imagen).

### 2.2.4.3. *Uso de convoluciones en redes*

¿Para qué se realiza este proceso? ¿Para qué queremos otra imagen? Este proceso se realiza para extraer características de la imagen usando kernels específicos, por ejemplo, se pueden crear imágenes que resaltan los bordes de los objetos.

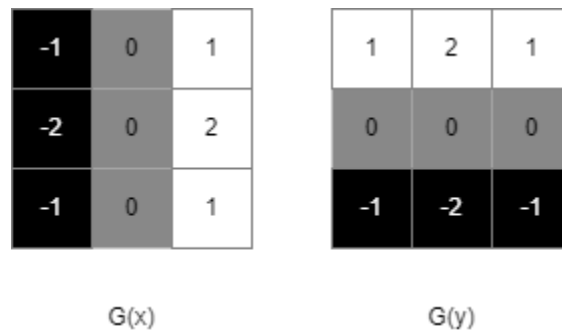


Figura 34 Figura de Operadores de Sobel

Como por ejemplo los kernels de la imagen anterior. Este tipo de kernels funciona de la siguiente manera:

Los kernels tienen una estructura que al convolucionar con la imagen original resta los valores de los píxeles de tal manera que si todos los valores de la imagen que se superponen con el kernel son iguales (no estamos cerca de un borde) la suma es cero, pero si existe un cambio de intensidad (borde) los valores serán positivos si cambiamos de una zona de baja intensidad a alta intensidad y negativos si pasamos de una zona de una zona de alta intensidad baja intensidad. Pero eso se entiende mejor con un ejemplo visual.



Figura 35 Imagen original



Figura 36 Imagen de Bordes horizontales



Figura 37 Imagen Bordes Sobel



Figura 38 Imagen bordes verticales

#### 2.2.4.4. *Uso de convoluciones en redes*

Entendiendo lo que es una convolución nos interesa conocer cómo se aplica a las redes neuronales.

Existe un ejemplo práctico proporcionado de simulación [42] el cual introducimos como entra desta imagen de un número tres dibujado a mano.

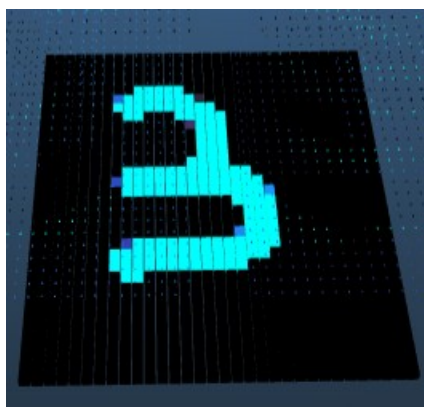


Figura 39 Imagen de número 3 dibujado a mano

A las siguientes imágenes se les aplican filtros (convoluciones) con la finalidad de extraer más información de la imagen. Convolucionando se obtienen las siguientes características:

Operación	Kernel	Imágen									
Bordes verticales con transición de zonas claras a zonas oscuras	<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	-1	-1	-1	0	0	0	1	1	1	
-1	-1	-1									
0	0	0									
1	1	1									
Entropía de la imagen											
Bordes diagonales 45°	<table border="1"> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>0</td><td>-1</td><td>-1</td></tr> </table>	1	1	0	1	0	-1	0	-1	-1	
1	1	0									
1	0	-1									
0	-1	-1									
Bordes diagonales -45°	<table border="1"> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>-1</td><td>0</td><td>1</td></tr> <tr><td>-1</td><td>-1</td><td>0</td></tr> </table>	0	1	1	-1	0	1	-1	-1	0	
0	1	1									
-1	0	1									
-1	-1	0									
Bordes horizontales con transición de zonas oscuras a zonas claras	<table border="1"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	1	0	-1	1	0	-1	
1	0	-1									
1	0	-1									
1	0	-1									
Bordes verticales con transición de zonas oscuras a zonas claras	<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	1	1	1	0	0	0	-1	-1	-1	
1	1	1									
0	0	0									
-1	-1	-1									

TABLA 1 IMÁGENES Y CONVOLUCIONES

#### 2.2.4.5. Redes neuronales residuales

Otro tipo de redes, también de aprendizaje profundo son las ResNet (Residual neural networks) o redes neuronales residuales. A esta familia pertenece la red que emplearemos en el proyecto, la RESNET50. Este tipo de redes se caracterizan por la utilización de “shortcuts”. Esto sucede cuando las entradas de una capa se salta el proceso de conexión tradicional y esa entrada se alimenta a una neurona más profunda dentro de la red. ¿Qué ventajas supone esta técnica? Este tipo de conexiones habilita el uso de redes mucho más

profundas que sin contar con ellas, ya que solían obtenerse peores resultados por un fenómeno denominado desvanecimiento de gradientes. Este proceso tenía lugar en el momento en el que muchas neuronas, el efecto de cada una con respecto a la salida es menor hasta llegar prácticamente a cero. Cada capa es la influencia por múltiples neuronas en la capa anterior, las cuales distribuyen la influencia. Si este proceso se repite infinitas veces, la influencia que tiene las capas tiende a cero, en cuyo caso es un dato imposible de mejorar.

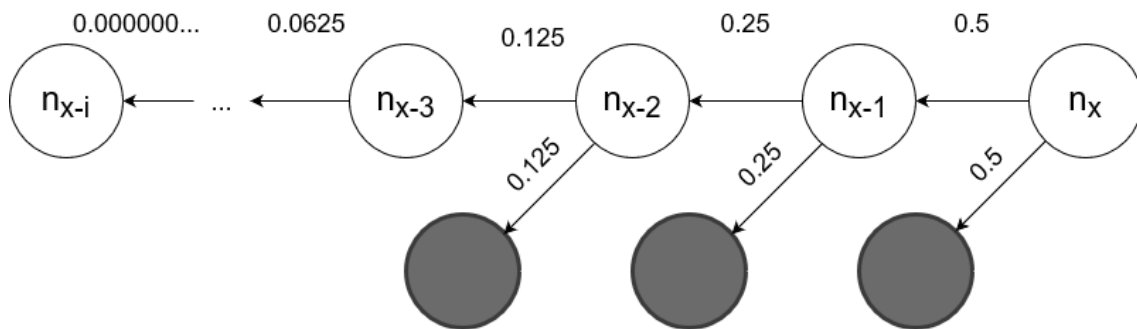


Figura 40 Ejemplo de la pérdida de influencia según la profundización en la red

Este suceso implica que ciertas neuronas pierdan la capacidad de auto-actualizarse, lo que perjudica el desempeño de la red. Es por ello que al integrar esta entrada en capas se mitiga el proceso de desvanecimiento de gradientes.

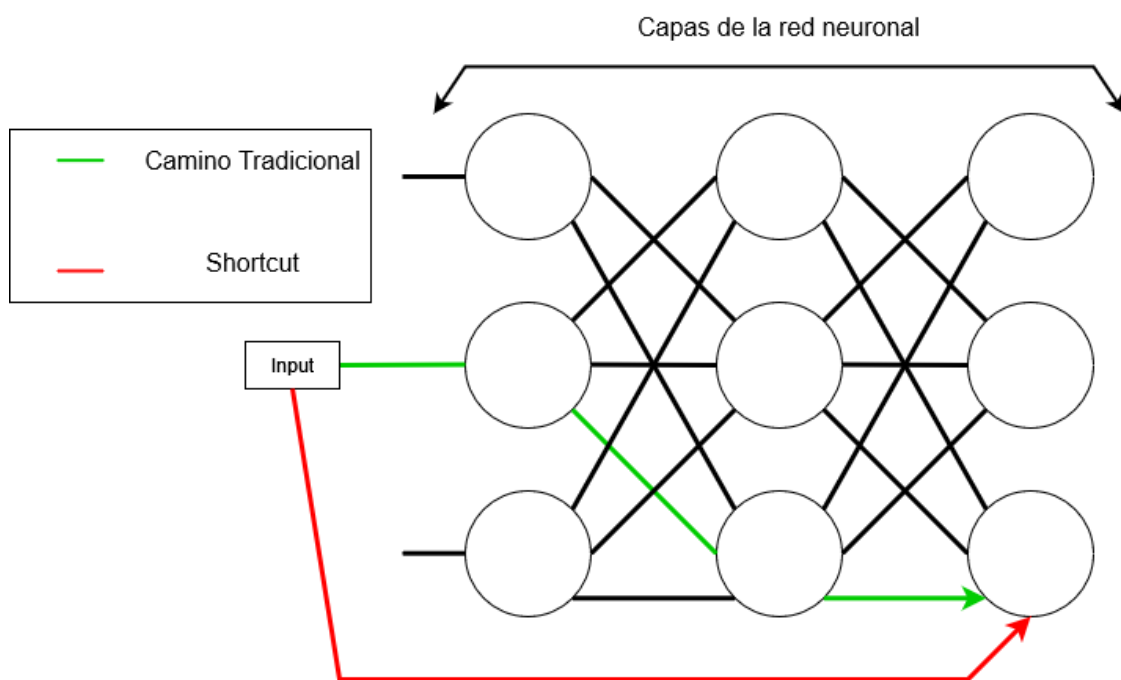


Figura 41 Proceso de shortcut o skipping

### 2.2.5. Transferencia de conocimiento

De entre todas las técnicas de Aprendizaje Máquina esta es tal vez la más relevante. Este es el método que emplearemos principalmente para realizar el entrenamiento de las redes neuronales.



¿En qué consiste esta técnica? La Transferencia del conocimiento [43] se basa en una idea simple, re-explotar el conocimiento ya adquirido por otras redes. A continuación, realizaremos distintas técnicas para ajustar redes ya entrenadas como pueden ser RESNET o DenseNet. Dependiendo de la situación el transfer learning se clasifica en tres tipos:

- **Aprendizaje por transferencia inductiva:** El campo de la fuente y el objetivo son los mismos, así mismo lo son los datos, en cambio las tareas de fuente y objetivo son diferentes, pero con ciertos elementos parecidos. La idea consiste entonces en usar los modelos existentes para reducir de manera ventajosa el campo de aplicación de los modelos posibles (sesgo del modelo).
- **Aprendizaje por transferencia no supervisada:** Como en el caso anterior, los campos de la fuente y el objetivo son similares pero las tareas son diferentes. Los datos en ambos campos no están etiquetados.
- **Aprendizaje por transferencia transinductiva:** Las tareas fuente y objetivo son similares pero los subcampos correspondientes son diferentes en términos de datos o de distribuciones de probabilidad marginales.

También existen diferentes maneras de aprovechar el conocimiento. Basándonos en cómo empleamos la red a reutilizar existen dos técnicas:

1. **Uso de modelos pre-entrenados como extractores de características:** La reutilización de una red de una capa final es la misión primordial. La nueva red funciona como un extractor de características fijas para realizar otras tareas. En estos modelos el entrenamiento se fundamenta en una nueva capa añadida al final, que será lo que permitirá adaptar nuevas características.
2. **Ajustes de modelos pre-entrenados:** En este método no se reemplaza la última capa, sino que se re-entrena todo el modelo. Esta estrategia, aunque más compleja, supone una ventaja dado que en lugar de entrenar una red desde cero se parte desde un punto más avanzado. Esto resulta en una optimización obteniendo mejores resultados en menos tiempo.

En nuestros ensayos utilizaremos el Aprendizaje por transferencia transinductiva para ajustar modelos pre-entrenados. Un nuevo conjunto de datos para reentrenar modelos.

## 2.3. Evaluación de datos

Seguidamente se explica cuáles fueron los métodos utilizados para la valoración de las redes. Se ilustran las técnicas de selección de datos y las métricas calculadas sobre los mismos.

### 2.3.1. Conjuntos de validación

Para el cálculo de estas métricas necesitamos designar un grupo de datos y apartarlo del entrenamiento. Esta acción se realiza para obtener una visión objetiva del desempeño de la red, por ende, se intenta no contaminar los datos del proceso de entrenamiento. Independientemente deben cumplirse otras propiedades como contar con un número de datos suficiente para tener importancia estadística (no menos de 1000 datos), ser un conjunto representativo de los datos reales (representaciones de los posibles escenarios) y debe ser escogido aleatoriamente de entre la totalidad de datos (la selección manual puede producir sesgos en la validación).

### 2.3.2. Validación cruzada

Se usan técnicas de validación cruzada como K-Fold o K-Fold estratificado. Esta familia de técnicas consiste en la división de los datos en varios conjuntos para el entrenamiento y el cálculo de métricas. La información proporcionada es de gran valor en lo relativo al estado de la red, indicando si esta sobreentrenada o su capacidad de generalización.

#### 2.3.2.1. K-Fold

Técnica consistente en dividir el conjunto de datos en  $k$  segmentos y realizar  $k$  entrenamientos donde se usa un segmento como conjunto de validación y el resto como datos para el entrenamiento. Así pues, se garantiza que los resultados serán objetivos independientemente de si una separación concreta es desfavorable para el entrenamiento.

La siguiente figura es un ejemplo de K-fold con 10 particiones. Por ejemplo, para el primer entrenamiento reservamos el conjunto 1 para realizar la validación sobre los datos y emplearemos los datos de los conjuntos 2-10 como datos de entrenamiento. Este proceso se repite  $n$  veces hasta que todos los conjuntos han sido usados al menos una vez como conjuntos de validación.

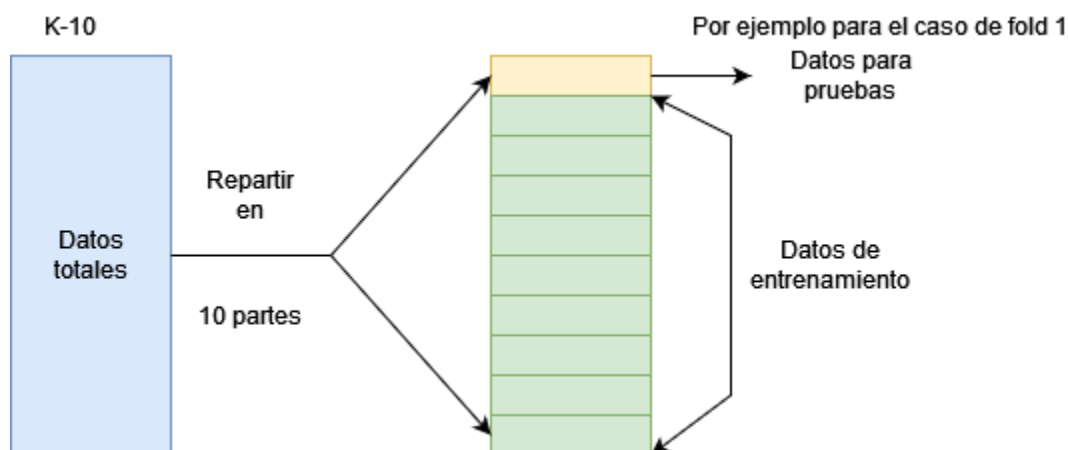


Figura 42 K-Fold en 10 conjuntos

#### 2.3.2.2. K-Fold Estratificado

Esta variación de K-Fold se caracteriza por asegurar que todos los conjuntos tienen la misma proporción de datos de las diferentes clases. Esto se realiza porque de lo contrario, podría ocurrir que debido a una mala partición queden conjuntos sin el número de datos necesarios en cada clase para la realización de un entrenamiento válido. Por medio de este proceso se garantiza que ninguno de los entrenamientos ha quedado sesgado por una mala distribución de los datos.

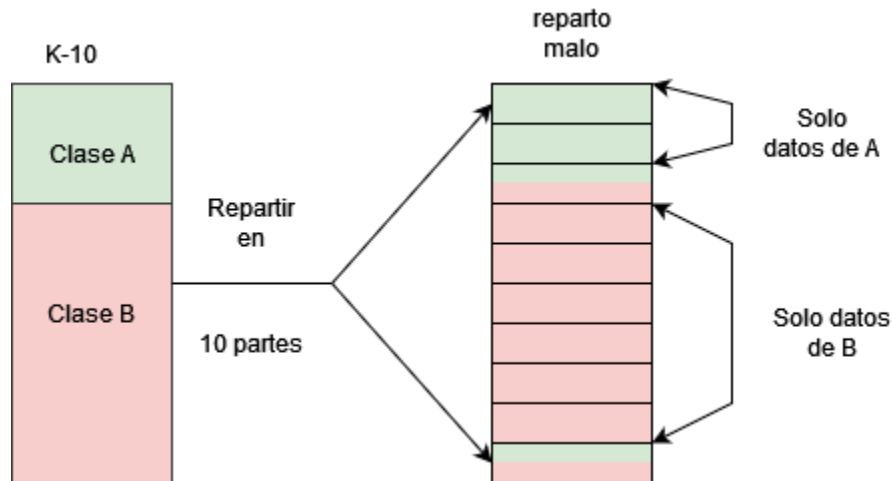


Figura 43 K-Fold con mala partición

### 2.3.3. Métricas de desempeño

Determinado el conjunto de datos para el cálculo de las métricas, debe determinarse cuáles de ellas se va a emplear. Existen diversas métricas de desempeño, pero en este estudio se decide emplear el conjunto de métricas multiclase [44].

#### 2.3.3.1. Métricas de desempeño Multiclase

Como la naturaleza del proyecto es multiclase se adoptan las métricas de clasificación binaria con la adaptación de “una contra todas” donde se escoge una clase  $C_i$  para actuar como la “positiva” y el resto como “negativas”. Esto se repite hasta obtener los resultados de “positiva” para todas las clases.

Métrica	Ecuación
Verdadero positivo $C_i$ Número de individuos pertenecientes a la clase $C_i$ que son clasificados correctamente	tp
Verdadero negativo $C_i$ Número de individuos pertenecientes a cualquier clase menos $C_i$ que son clasificados correctamente	tn
Falso positivo $C_i$ Número de individuos no pertenecientes a $C_i$ que son clasificados como $C_i$	fp
Falso negativo $C_i$ Número pertenecientes a $C_i$ que son clasificados en otra clase	fn

TABLA 2 VALORES DE LAS ECUACIONES

Métrica	Ecuación
<p>Accuracy (Exactitud) <math>C_i</math></p> <p>Medida que determina la eficacia general de un clasificador y que es calculada como la suma total de los casos clasificados correctamente (verdaderos positivos, verdaderos negativos) divididos por el número de casos totales (verdaderos positivos, falsos positivos, verdaderos negativos)</p>	$\frac{tp + tn}{tp + fp + tn + fn}$
<p>Precision (Precisión) <math>C_i</math></p> <p>Métrica que evalúa qué porcentaje de elementos que clasificamos pertenecen realmente a la clase. Esto se calcula como los clasificados correctamente <math>tp</math> entre el total de los realmente pertenecientes a la en esa clase (<math>tp + fp</math>)</p>	$\frac{tp}{tp + fp}$
<p>Recall (Exhaustividad) <math>C_i</math></p> <p>Métrica que evalúa la capacidad que tiene el clasificador de detectar el número positivos para la clase <math>C_i</math> entre los existentes (que porcentaje de elementos de la clase quedan sin detectar) esto se calcula como los clasificados correctamente <math>tp</math> entre el total de los realmente pertenecientes en esa clase (<math>tp + fn</math>)</p>	$\frac{tp}{tp + fn}$
<p>Specificity (Especificidad) <math>C_i</math></p> <p>Métrica que evalúa la capacidad que tiene el clasificador de detectar los verdaderos negativos <math>C_i</math> esto se calcula como los clasificados correctamente <math>tn</math> entre el total de los clasificados en las otras clases (<math>fp + tn</math>)</p>	$\frac{tn}{fp + tn}$
<p>F1-score <math>C_i</math></p> <p>media armónica entre precision y recall usando el valor de <math>\beta</math> para ponderar qué métrica es más importante.</p>	$\frac{(\beta^2 + 1)tp}{(\beta^2 + 1)tp + \beta^2 fn}$
<p>AUC</p>	$\frac{1}{2} \left( \frac{tp}{tp + fn} + \frac{tn}{tn + fp} \right)$

TABLA 3 MÉTRICAS DE DESEMPEÑO MULTICLASE

Estimando la naturaleza multiclase del proyecto se adoptan métricas de clasificación binaria con la adaptación de “una contra todas”.

### 2.3.3.2. Métricas Generales

Tras calcular las métricas para cada clase se calcula el promedio entre ellas para la evaluación general de la red.

Métrica	Wcuación
Media aritmética de la exactitud individual de cada una de las clases	$\sum_{i=1}^l \frac{Acc_{C_i}}{l} = \frac{\sum_{i=1}^l \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$
Error rate Grado de error en la predicción respecto a los datos verdaderos.	$\frac{\sum_{i=1}^l \frac{fp_i+fn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$
Precision $\mu$ Media de la precisión de todas las clases calculada con la suma de los valores individuales de cada clase.	$\frac{\sum_{i=1}^l tp_i}{\sum_{i=1}^l (tp_i+fn_i)}$
Recall $\mu$ Media de la recall de todas las clases calculada con la suma de los valores individuales de cada clase.	$\frac{\sum_{i=1}^l tp_i}{\sum_{i=1}^l (tp_i+fp_i)}$
Fscore $\mu$ media armónica entre precision y recall usando el valor de $\beta$ para ponderar qué métrica es más importante.	$\frac{(\beta^2+1)Precision_{\mu}Recall_{\mu}}{\beta^2Precision_{\mu}+Recall_{\mu}}$
Precision M Media aritmética del precision de todas las clases	$\frac{\sum_{i=1}^l \frac{tp_i}{tp_i+fn_i}}{l}$
Recall M Media aritmética del recall de todas las clases	$\frac{\sum_{i=1}^l \frac{tp_i}{tp_i+fp_i}}{l}$
F-score Se calcula igual que Fscore $\mu$ , reemplazando la precisión $\mu$ y exhaustividad $\mu$ por precisión M y exhaustividad M	$\frac{(\beta^2+1)Precisión_MRecall_M}{\beta^2Precision_M+Recall_M}$

TABLA 4 MÉTRICAS DE DESEMPEÑO GENERALES

### 3. Metodología

Procederemos a explicar brevemente la planificación del proyecto así como la justificación de la metodología, planificación y técnicas empleadas.

#### 3.1. Justificación metodológica

Para este proyecto usé la metodología de planificación espiral. Ya que está diseñada para un desarrollo iterativo que permite el refinado progresivo de prototipos. Lo que se adapta a mejor a un proyecto de investigación que las metodologías basadas en producto, como las de desarrollo ágil.

#### 3.2. Planificación Espiral

“El desarrollo en espiral es una familia de procesos de desarrollo de software caracterizada por la repetición. Iterando rápidamente un conjunto de procesos de desarrollo elemental y gestionando el riesgo para que sea activamente reducido” [45].

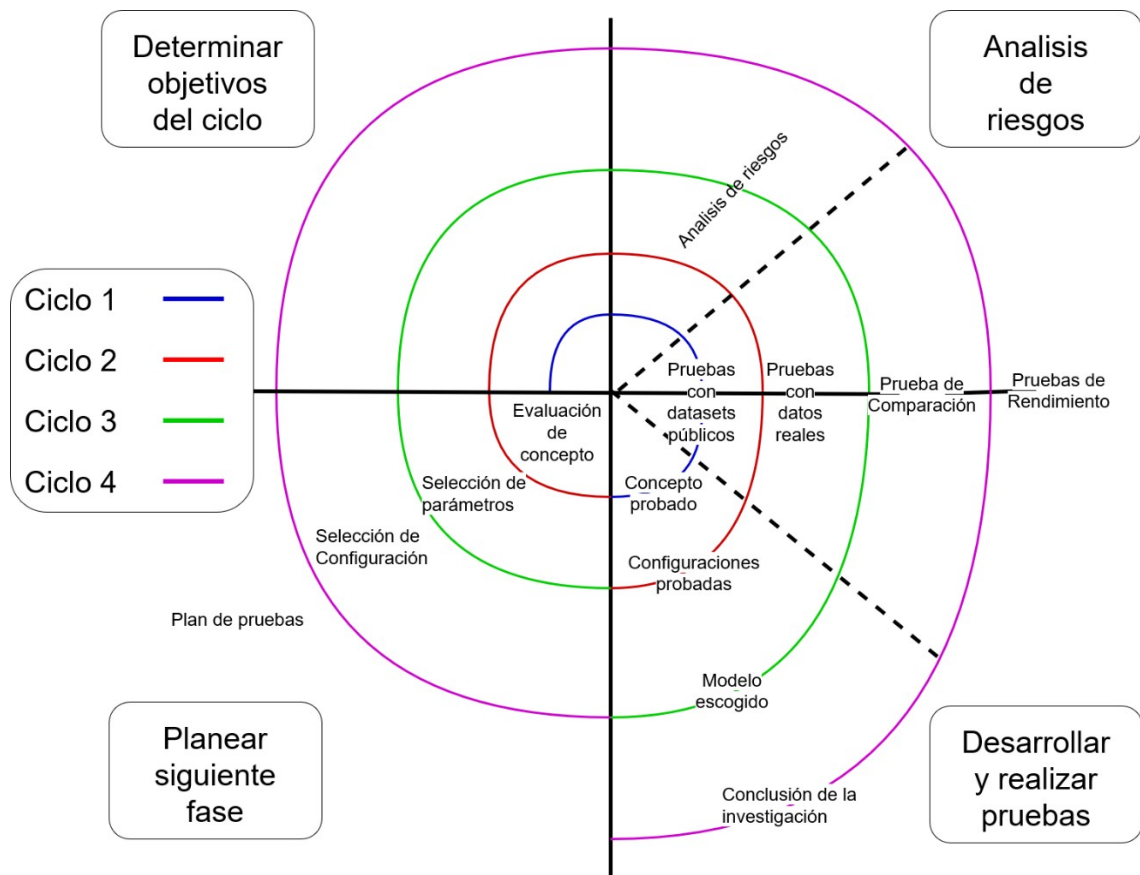


Figura 44 Diagrama de desarrollo en espiral

El desarrollo de cada ciclo consiste en cuatro fases:

- a) Propuesta de objetivos: Para comenzar hay que clarificar los objetivos que se esperan obtener durante el ciclo. Esto no incluye únicamente los requerimientos técnicos del programa, sino también los recursos y la temporalización de su desarrollo.
- b) Análisis de riesgo: Fase de indentificación y analisis de los posibles riesgos que pueden afectar al desarrollo del proyecto. Se planifican contingencias para prevenirlos o afrontarlos.
- c) Desarrollo y testing: Etapa de realización del desarrollo y pruebas del programa conforme a los objetivos establecidos.

### **3.3. Planificación**

He diseñado cuatro ciclos en el desarrollo<sup>2</sup>. Para cada ciclo, como hemos visto existen cuatro fases. 2 fases de planificación, una fase de ejecución y una fase de preparación. A continuación resumiremos los elementos fundamentales de las fases de planificación de cada ciclo omitiendo la fase de preparación para el siguiente ciclo por razones de simplicidad.

#### **3.3.1. Ciclo de apertura**

El propósito de esta etapa es llevar a cabo experimentos e investigaciones con el objetivo de ampliar nuestro conocimiento con respecto al Aprendizaje Automático relacionado con técnicas y metodologías específicas.

Si bien los resultados de las pruebas realizadas durante esta etapa no contribuyen al producto final de nuestra investigación, pretendemos que las técnicas y conocimientos adquiridos cobren relevancia en los ciclos ulteriores.

Aunque aún no sabemos cual es la profundidad de los conocimientos y técnicas que debemos adquirir siendo muy generosos dedicaremos a la investigación y pruebas de esta fase de entre un mes y medio y dos meses.

##### **3.3.1.1. Objetivos del ciclo**

Probar diferentes técnicas de aprendizaje automático que van desde un simple RNN de arquitectura hecha por nosotros hasta la reutilización de redes preestablecidas a través del aprendizaje por transferencia.

Usar conjuntos de datos pre-probados como Cifar 10 con el objetivo de hacer que el análisis de las técnicas y redes sea lo más fácil y objetivo posible.

Encontrar un conjunto de métricas estándar y técnicas de evaluación que proporcionen un estándar claro y resultados comprensibles. Evaluar diferentes casos con las métricas. Planeamos probar conjuntos de datos binarios y multiclase con múltiples RNN con el objetivo de comprender las diferencias y los enfoques necesarios para cada caso.

##### **3.3.1.2. Análisis de riesgos**

Inherentemente de todas las fases del proyecto esta es tal vez la que más riesgos puede conllevar dado el grado de desconocimiento del problema y la falta de experiencia previa dependeremos de la ayuda del mentor para el desarrollo de esta fase.

##### **3.3.1.3. Desarrollo y realización de pruebas**

---

<sup>2</sup> Los ciclos que se realizan de forma iterativa son la definición, el entrenamiento y la validación de modelos.



Se realizarán ensayos preliminares para familiarizarse con las técnicas empleadas, y las librerías utilizadas. Usando datasets públicos.

Con el objetivo de familiarizarme con el sistema y los módulos de Python para el entrenamiento de Redes Neuronales, realicé diversas pruebas con un dataset conocido como CIFAR-10

### **3.3.2. Ciclo 2: Introducción a los datos reales**

En este ciclo se incorporan los datos reales del problema al ciclo de desarrollo. Empezaremos de forma sencilla, utilizando únicamente dos de las dieciséis ratas del experimento. Seleccionaremos un par de modelos de los modelos disponibles que ofrece el paquete Keras para probar diferentes configuraciones y sus impactos.

Planeamos asignar a esta sección del proyecto otros dos meses. El cronograma consistirá en tres semanas de codificación de software seguidas de cinco semanas de entrenamiento de los modelos, pruebas y correcciones menores del código.

#### **3.3.2.1. Objetivos del ciclo**

Encontrar un conjunto de modelos para realizar las pruebas.

Definir una serie de ensayos para evaluar el impacto en el rendimiento que tienen ciertas variables sobre el rendimiento de los modelos.

#### **3.3.2.2. Análisis de riesgos**

El mayor riesgo al que se enfrenta esta sección son los errores menores y las inconsistencias que arruinan la base estadística que planeamos proporcionar con este proyecto. Por lo tanto, ponemos especial atención en la revisión

#### **3.3.2.3. Desarrollo y realización de pruebas**

Al igual que al inicio simplificamos el problema probando con una tarea más sencilla (CIFAR-10), se procede del mismo modo a menos escala con los datos reales. En lugar de ensayar con todas las ratas de laboratorio, se comienza únicamente con dos individuos. La finalidad es incrementar la claridad de los resultados y simplificar la resolución del problema.

### **3.3.3. Ciclo 3: Prueba de comparación de modelos**

Se ponen en práctica los modelos seleccionados y se escoge un modelo singular para la etapa de cierre.

Gracias a los pequeños cambios que deben realizarse, planeamos completar este ciclo en el corto período de tiempo de dos semanas o una semana y media, todo esto dedicado a entrenar los modelos con, con suerte, cambios mínimos en el código del proyecto.

#### **3.3.3.1. Objetivos del ciclo**

El objetivo fundamental de este ciclo es obtener una valoración final del rendimiento de los diferentes modelos en el problema. Y la selección del mejor modelo.

#### **3.3.3.2. Análisis de riesgos.**

No se han detectado riesgos específicos en este ciclo.

#### **3.3.3.3. Desarrollo y realización de pruebas**

Con anterioridad a la realización de las pruebas finales, se realiza un ensayo intermedio con dieciseis ratas de laboratorio para comparar el desempeño de los modelos. Los objetivos son escoger varios modelos, entrenarlos todos con los mismos parámetros y seleccionar el mejor de ellos.

#### **3.3.4. Ciclo de cierre: Análisis del rendimiento de la red**

Seleccionado el mejor modelo para resolver el problema, en este ciclo someteremos a prueba el modelo seleccionado para comprobar su habilidad para identificar animales en diferentes situaciones.

##### **3.3.4.1. *Objetivos del ciclo***

Realizar una validación robusta del modelo en diferentes escenarios.

Encontrar los límites del modelo para resolver el problema. Por ejemplo, el número máximo de individuos y número mínimo de ejemplos por individuo que necesita el modelo para realizar una identificación robusta de los animales.

Poner a prueba la capacidad de generalización del modelo añadiendo variaciones en los datos de test.

##### **3.3.4.2. *Análisis de riesgos***

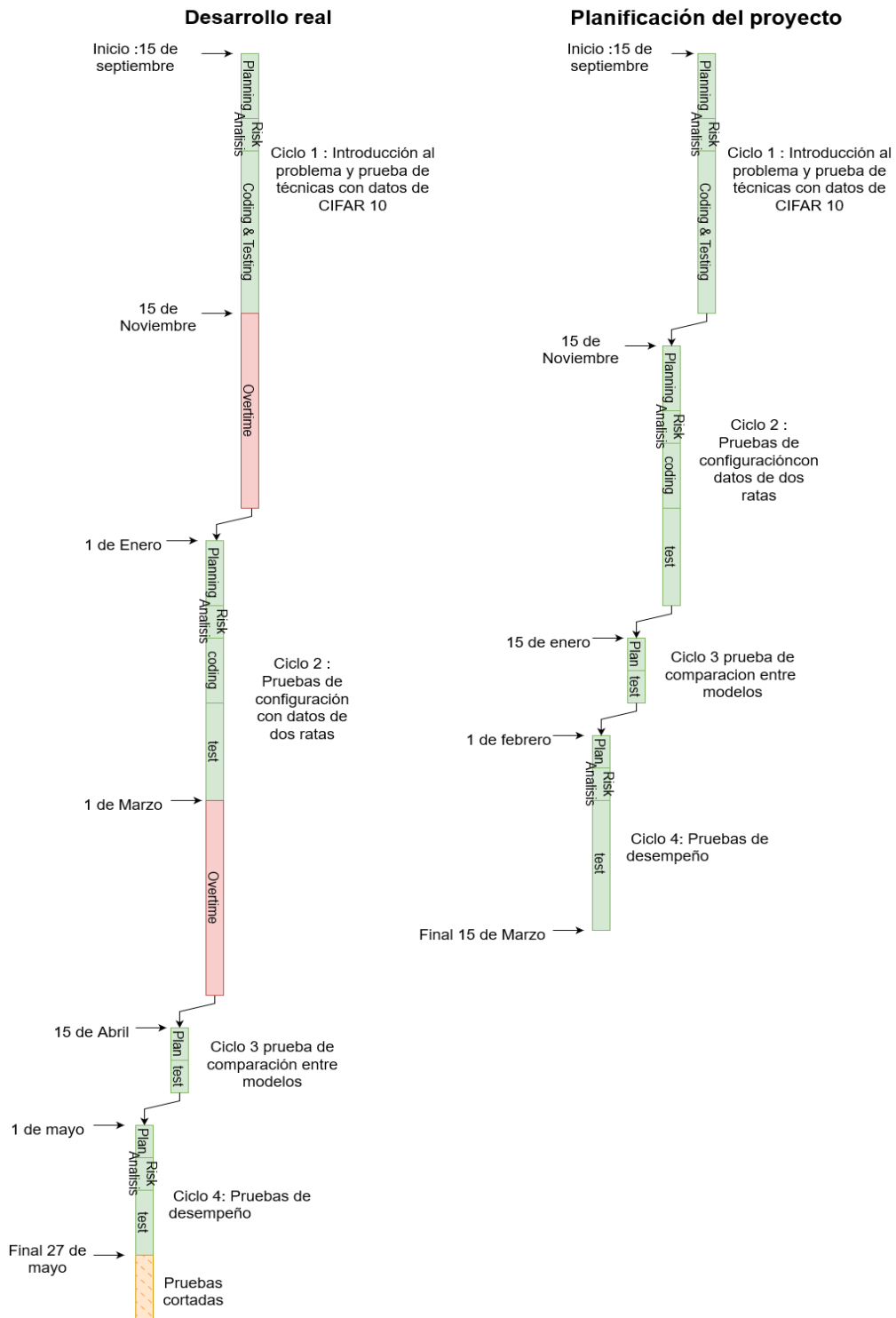
El principal riesgo que implica esta fase surge del hecho de que estaremos cambiando y modificando los conjuntos de datos una vez más, lo que requiere una atención especial para no contaminar los resultados del entrenamiento con malos resultados de procesamiento. Por lo tanto, haremos un seguimiento con el tutor y revisaremos de cerca el código para garantizar la integridad del proyecto.

##### **3.3.4.3. *Desarrollo y realización de pruebas***

Por último, para probar el desempeño de la red se realiza un ensayo para determinar la relación entre datos de entrenamiento y número de animales que pueden clasificarse de forma fiable. La prueba consistió variar de individuos (clases) a identificar, y al mismo tiempo probar el rendimiento de la red alterando la cantidad de datos usados para cada clase durante el entrenamiento.

### 3.4. Evaluación del desarrollo del proyecto

Figura 45



Planificación inicial del proyecto vs desarrollo real

## 4. Desarrollo

**D**espués de explicar cómo se ha desarrollado la fase de planificación procedemos a la sección de desarrollo. En ella explicaré cuales han sido los pasos realizados durante el proyecto.

### 4.1. Entorno software

Previo al desarrollo de la investigación es necesaria la instalación de las aplicaciones que posibiliten su desempeño como son el intérprete, las librerías y las aplicaciones de soporte.

#### 4.1.1. Descarga del intérprete

Para el desarrollo del Proyecto Python ha sido el lenguaje elegido, lo que implica el uso del intérprete correspondiente. Esta aplicación permite ejecutar el código humano a código máquina. Finalmente, me decanté por la versión 2.8.10 ya que no conlleva problemas de compatibilidad con una amplia gama de galerías,

#### 4.1.2. Librerías

A continuación, expongo la relación de librerías empleadas:

##### 4.1.2.1. Tensorflow

Librería fuente de la mayor parte de problemas de compatibilidad del entorno dado que debe ser compatible con los controladores de la gráfica y las librerías CUDA y Numpy.

##### 4.1.2.2. Cudnn y CUDA

Es recomendable instalar Cudnn y CUDA por ser librerías que posibilitan el uso de GPU necesaria para acelerar significativamente (o por lo menos 10 veces más rápido) la ejecución de las redes neuronales.<sup>3</sup>

##### 4.1.2.3. Numpy con Matplotlib y Pandas

También debemos contar con versiones compatibles de Numpy, Matplotlib y Pandas.

##### 4.1.2.4. Skimage v0.19-2 Opencv

Otras librerías a mi disposición han sido Skimage V0.19.3 y Opencv, dado que se dedican al tratamiento de imágenes con el objetivo de realizar operaciones aumento de datos y visión artificial. Se emplearán para el entrenamiento de las redes neuronales con la finalidad de optimizar su desempeño.

---

<sup>3</sup> Las cuales deben ser 8.1 y 11.2 respectivamente para resultar compatibles con la versión de Tensorflow.

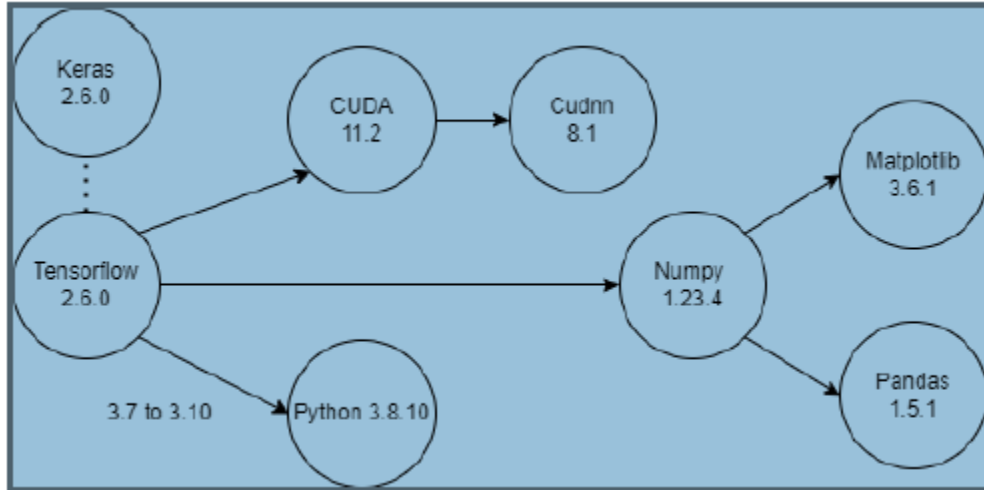


Figura 46 Cascada de dependencias entre librerías y programas

### 4.1.3. Otros recursos

#### 4.1.3.1. Visual Studio 2017

Considero Visual Studio 2017 lo más recomendable dadas las funcionalidades adicionales de éste software, considerando el entorno, y a recomendación de los desarrolladores CUDA y Cudnn. Esta es la última versión compatible<sup>4</sup> con la versión de CUDA instalada [46].

#### 4.1.3.2. Geany

Por preferencia personal instalé Geany. No ofrece un sistema de organización de Proyecto como Visual Studio, pero sí un entorno básico de programación para la ejecución y compilación de los programas directamente desde la aplicación.

<sup>4</sup> Fue necesario recuperarla de un repositorio (ahora Microsoft provee de un acceso más sencillo a estas versiones [47], dado que por defecto solo accedemos a la versión actualizada. La versión de 2020 causaba problemas e la ejecución del código.

## 4.2. Desarrollo del proyecto

Esta parte de la sección de desarrollo está dedicada a explicar qué se ha realizado durante el desarrollo del proyecto, así como la justificación teórica de las decisiones tomadas. Según la motivación, las pruebas realizadas y los datos empleados, la investigación puede dividirse en cuatro partes: Una breve fase de familiarización con el entorno donde se prueban diferentes técnicas y modelos contra una base de datos estándar. A continuación, una siguiente fase en la que ya se trabaja con datos reales probando los diferentes modelos y configuraciones. El objetivo era la mejora de resultados. Para continuar una breve fase comparativa de modelos con una configuración básica, para la selección final de uno de ellos. Finalmente, se concluye con una prueba de reacción del modelo a distintos números de clases y datos.

### 4.2.1. Pre-proyecto y prueba de concepto con dataset CIFAR-10

Con el objetivo de familiarizarme con el sistema y los módulos de Python para el entrenamiento de Redes Neuronales, realicé diversas pruebas con un dataset conocido como CIFAR-10: KNN-Based Ensemble of Classifiers [48]. Me resultó la herramienta más apropiada para mostrar empíricamente la mejora del rendimiento en la combinación de clasificadores CNN con KNN o CIFAR-10 <sup>5</sup>Image Classification with Convolutional Neural Networks for Embedded Systems [49]. Este conjunto de datos consiste en imágenes de 32x32 píxeles de resolución pertenecientes a 10 clases distintas: aviones, barcos, caballos, camiones, ciervos, coches, gatos, pájaros perros y ranas.

Las decisiones determinantes en esta fase fueron la selección de las redes de experimentación y el desbloqueo de los pesos. No se realizaron experimentos con todas las redes disponibles de Keras por tres razones:

- No todas las redes se adecuaban al objetivo.
- Muchas de las redes son simplemente diferentes versiones <sup>6</sup>de las mismas.
- Presentan otros problemas. ConvNeXtXLarge es demasiado pesado para entrenar con el equipo con el que se realizaron las pruebas.

La determinación que causó mayor impacto sobre el rendimiento de las redes fue el descarte del conocimiento previo. Normalmente cuando se utiliza un Aprendizaje de Transferencia, ciertas capas de la red quedan bloqueadas para no destruir el conocimiento ya adquirido. De este modo, la función de Transferencia de Aprendizaje se convierte en un proceso de ajuste. En mi caso, ya que las redes tienen un único objetivo, se probó a eliminar esas restricciones. Esto mejoró en gran medida los resultados de la red (de un 60 % a casi un 100 % de precisión). Por este motivo, este es el proceso que se realizará en todas las pruebas posteriores.

### 4.2.2. Ensayo de Exploración de Parámetros

Comprendido el entrenamiento de las redes neuronales y los parámetros que afectan a los resultados, se procede a la fase de pruebas con datos reales. Se comienza a hacer uso de las imágenes de las ratas de laboratorio preparadas con anterioridad para proceder con el entrenamiento.

Al igual que al inicio simplificamos el problema probando con una tarea más sencilla (CIFAR-10), se procede del mismo modo a menos escala con los datos reales. En lugar de

---

<sup>5</sup> Se muestra el rendimiento de sistema con memoria inferior a 2GB para así poder incluirla en sistemas embebidos.

<sup>6</sup> EfficientNetB1, EfficientNetB2,... EfficientNetB7.

ensayar con todas las ratas de laboratorio, se comienza únicamente con dos individuos. La finalidad es incrementar la claridad de los resultados y simplificar la resolución del problema<sup>7</sup>.

Dentro de esta fase se realizaron las siguientes pruebas: Pruebas con diferentes ciclos de entrenamiento, pruebas con diversos tamaños de imagen, pruebas alterando el tamaño de muestra y pruebas con distintas métricas para computar la pérdida.

#### 4.2.2.1. *Epochs o ciclos de entrenamiento*

La primera prueba que se realiza es el efecto que tiene el número de ciclos de entrenamiento o Epochs sobre el desempeño de en las siguientes redes: RESNET-50, MobileNet V2, DesNet 201 y EfficientNet-B7. Se escogen estas cuatro redes de entre las usadas anteriormente en la fase de pruebas como representante de su familia de redes.

#### 4.2.2.2. *Tamaño de imagen*

A continuación se realizan pruebas alterando el tamaño de la imagen, el efecto que debe variar el tamaño<sup>8</sup> de las imágenes en el tiempo de entrenamiento y el desempeño de los modelos<sup>9</sup>.

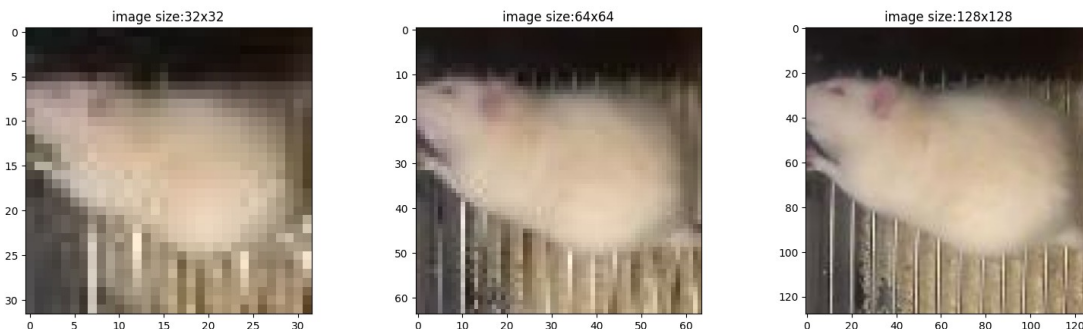


Figura 47 Imágenes de las ratas de laboratorio en los tres tamaños ensayados

#### 4.2.2.3. *Tamaño de la muestra*

La penúltima prueba que se realiza es observar el efecto que tiene cambiar el tamaño de muestra (número de muestras que con las que se entrena la red antes de actualizar los pesos) sobre el funcionamiento de las redes. Al igual que en la prueba anterior debido a la diferencia de tamaño entre las redes no todos los tamaños de muestra se pueden probar con todas las redes es por eso que los tamaños máximos de prueba con respecto a las redes fueron MobileNet\_V2 : 1024, RESNET\_50 : 512, DenseNet\_201 : 256 y EfficientNet\_B7 64.

Se exponen los datos del tamaño de la muestra en el apartado 6 de resultados

#### 4.2.3. **Prueba de comparación de modelos**

Con anterioridad a la realización de las pruebas finales, se realiza un ensayo intermedio con dieciseis ratas de laboratorio para comparar el desempeño de los modelos. Los objetivos son

<sup>7</sup> En lo que refiere al tamaño de la imagen, los primeros datos que se intentan cargar al sistema de prueba sobrecargan la memoria. La evidencia de sobrecarga por la magnitud de la imagen descarta la sobrecarga por el tamaño de los datos.

<sup>8</sup> Con imágenes de 128 sólo se entrenará RESNET-50 y MobileNet V2 porque Efficient Net y DenseNet son desmesurados para que el equipo de prueba pueda cargar las redes y las imágenes 128x128 simultáneamente.

<sup>9</sup> Se entrena con imágenes de 32x32 píxeles, 64x64 y 128x128.

escoger varios modelos, entrenarlos todos con los mismos parámetros<sup>10</sup> y seleccionar el mejor de ellos.

#### **4.2.4. Análisis del rendimiento de la red**

Por último, para probar el desempeño de la red se realiza un ensayo para determinar la relación entre datos de entrenamiento y número de animales que pueden clasificarse de forma fiable. La prueba consistió variar de individuos (clases) a identificar, y al mismo tiempo probar el rendimiento de la red alterando la cantidad de datos usados para cada clase durante el entrenamiento.

La validación de este ensayo se realizó separando mil datos aleatorios por individuo para el conjunto de test.

#### **4.2.5. Pruebas con imágenes en blanco y negro**

Por último, se realizó la prueba con individuos en blanco y negro, donde mediante el módulo `scikit imagen` se elimina el color de las imágenes, transformándolas a blanco y negro. Con este nuevo conjunto de datos se realizaron los entrenamientos nuevamente.

---

<sup>10</sup> Tamaño de muestra 32, tamaño de imagen 32x32 y función de pérdida *categorical cross entropy*.



## 5. Resultados

Esta sección está dedicada al análisis de resultados obtenidos durante el desarrollo del proyecto. Los datos presentados en este apartado son solo un subconjunto de los mismos. Esta sección está dividida según las fases de desarrollo del proyecto. De esta forma quedan excluidas las pruebas preliminares y se mostrarán solo las pruebas con los datos reales del proyecto.

### 5.1. Ensayo de Exploración de Parámetros

A continuación, expongo los resultados de las pruebas iniciales para probar diferentes composiciones con distintos modelos para poder compararlos. Durante este proyecto se han probado cuatro parámetros diferentes para el desempeño de las redes: Ciclos de Entrenamiento o Epochs, Tamaño de Imagen y tamaño de la muestra.

#### 5.1.1. Epochs

En este apartado se muestran los resultados de una prueba inicial donde se estudia el efecto que tienen el número de ciclos de entrenamiento sobre las redes

Tabla 5 Matriz de confusión Epochs RESNET-

	Rata 1	Rata 2
Rata 1	0.94808	0.00358
Rata 2	0.05192	0.99642

Tabla 6 Positivos u negativos Epochs red RESNET-50

	TP	TN	FP	FP
Rata 1	9480.8	9964.2	35.8	519.2
Rata 2	9964.2	9480.8	519.2	35.8

	recall	precision	sensitivity	specificity	PPV	NPV	f1
RESNET_50	0.97	0.97	97.22	0.97	97.90	97.22	0.97
MobileNet	0.99	0.99	99.87	0.99	99.87	99.87	0.99
DenseNet	0.99	0.99	99.96	0.99	99.96	99.96	0.99
EfficientNet	0.99	0.99	99.99	0.99	99.99	99.99	1.00

TABLA 7 MÉTRICAS DEL CASO DE EPOCHS MACRO-PROMEDIADA

#### 5.1.2. Tamaño de la Imagen

La siguiente prueba ensayada fue alterar el tamaño de la imagen. En experimentos anteriores se realizaron con ejemplos más simples como el de “Perseveración de la identidad de cada animal del total de las ratas de laboratorio en ensayos de comportamiento social” [50]. Los resultados sugieren que el aumentar el tamaño de las imágenes no tiene gran efecto, y de haberlo supone un minúsculo empeoramiento de los resultados.

	recall	precision	sensitivity	specificity	PPV	NPV	f1
RESNET	0.99	0.99	99.95	0.99	99.95	99.954	0.99
MobileNet	0.99	0.99	99.87	0.99	99.87	99.87	0.99
DenseNet	0.99	0.99	99.94	0.99	99.94	99.94	0.99
EfficientNet	0.99	0.99	99.96	0.99	99.96	99.96	0.99

TABLA 8 MÉTRICAS DEL CASO DE TAMAÑO DE IMAGEN 32X32 MACRO-PROMEDIADAS

En los ensayos de variación de Tamaño de la Imagen se puede observar una mejora directamente proporcional entre el aumento de la imagen y la calidad y estabilidad de los resultados. Sin embargo, a partir de cierto punto las mejoras<sup>11</sup> son casi imperceptibles. Si consideramos el incremento del tiempo de ejecución que implica el aumento de dimensión de las imágenes, el cual como mínimo, se dobla en cada muestra del ensayo, la mejora marginal no justifica el incremento de tiempo y recursos necesarios. Es por ello que en sucesivas pruebas se usarán como tamaño estándar imágenes de 32x32 píxeles

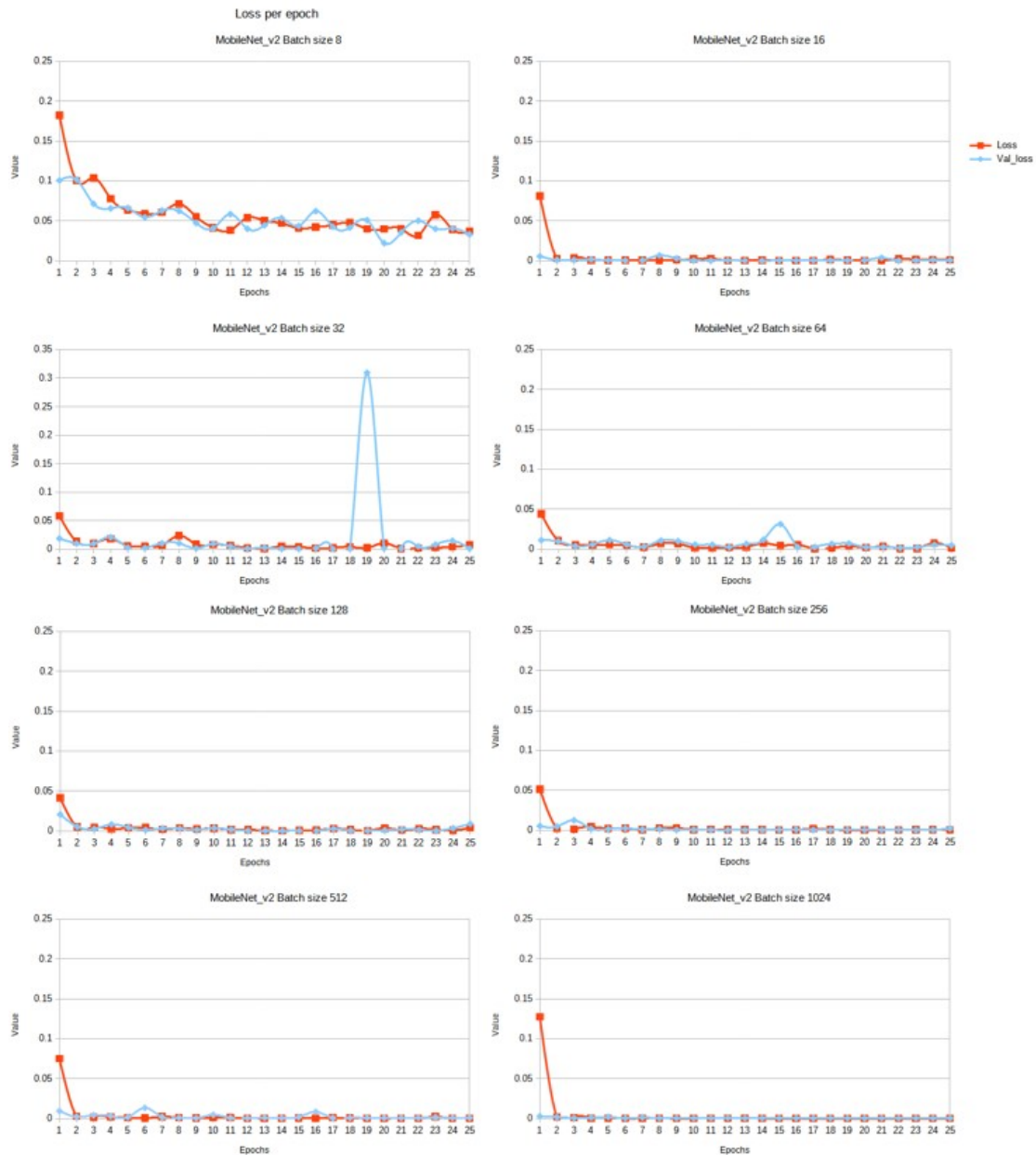
### 5.1.3. Tamaño de la muestra

En este ensayo se modifica el Tamaño de Muestra (nº de imágenes empleadas simultáneamente en cada ciclo de entrenamiento de la red). Esta prueba es importante ya que es uno de los parámetros clave que afecta al rendimiento y al tiempo de entrenamiento de la red.

tamaño de muestra	recall	precision	sensitivity	specificity	PPV	NPV	f1
8	0.99	0.99	99.64	0.99	99.64	99.64	0.99
16	0.99	0.99	99.49	0.99	99.49	99.49	0.99
32	1.00	1.00	99.98	1.00	99.98	99.98	1.00
64	0.99	0.99	99.87	0.99	99.87	99.87	0.99
128	0.99	0.99	99.55	0.99	99.56	99.55	0.99
256	0.99	0.99	99.64	0.99	99.64	99.64	0.99
512	0.99	0.99	99.58	0.99	99.58	99.58	0.99
1024	0.99	0.99	99.52	0.99	99.52	99.52	0.99

TABLA 9 MÉTRICAS POR TAMAÑO DE LA MUESTRA PARA EL MODELO MOBILENET-V2

<sup>11</sup> En el peor de los casos la precisión es de un 99.85 %.



**Figura 48** Ejemplo de valores de pérdida de la red MobileNet para distintos tipos de muestra

Como norma general, las redes neuronales suelen conservar mejores resultados y demorarse menos cuanto más grande es el tamaño de la muestra. Esto se debe a que cuando existe un aumento de la magnitud de la muestra, menos actualización precisa la red. Esto agiliza y aumenta la precisión al estar considerando un rango más amplio de datos.

Como inconveniente, destacar que un tamaño de muestra de mayor magnitud implica un consumo de memoria del parámetro de muestra muy superior.

Con estas conclusiones debo justificar el tamaño de mi muestra compuesta únicamente por 32 individuos. Aunque he determinado que un mayor número de muestra optimiza los resultados obtenidos por las redes neuronales, en mi ensayo no ha sido posible incrementarla por una cuestión de recursos, en concreto de memoria de software y procesadores.

## 5.2. Prueba de comparación de modelos

En este ensayo, realizamos una comparativa con la totalidad de los animales para los mejores parámetros (seleccionados en el ensayo anterior). Para los distintos modelos. Así la configuración usada para todas las técnicas es una resolución de imagen de 32x32 píxeles, un tamaño de 32 muestras y *sparse categorical cross entropy* como función de pérdida.

Cuando se trata de comparar el rendimiento de los cuatro modelos, ya que todos exhiben una gran precisión<sup>12</sup>, mi criterio se justifica en otros aspectos determinantes para el proyecto como el tiempo y tamaño de la red.

En base a estos factores, el modelo de menor tamaño, aproximadamente 40 MB y el tiempo de entrenamiento más rápido, de 55 segundos por Epoch es MobileNet-V2, son los seleccionados.

## 5.3. Análisis del rendimiento de la red

Durante esta última fase, se probó el desempeño de la red conforme a dos variables, en número de animales a identificar y número de ejemplos o imágenes de cada individuo utilizados para el entrenamiento. En esta prueba probamos a cambiar el número de clases y el número de imágenes de entrenamiento de la red.

Comprobamos que la red es bastante resistente a la reducción en los datos de entrenamiento puesto que puede identificar a las ratas con una certeza determinada del 95 % con tan solo 280 imágenes de ensayo por individuo. Especulo que esto se debe en gran medida a que el punto de partida no es una red vacía en su totalidad. Dado que estamos utilizando redes de reconocimiento de imágenes ya entrenadas y con un considerable rendimiento probado, estos resultados son satisfactorios.

Aunque no en el mismo campo, otras investigaciones [51], han manifestado conclusiones igualmente positivas del 99 % de precisión después del entrenamiento con un conjunto de datos reducido de únicamente 200 instancias.

## 5.4. Pruebas con imágenes en blanco y negro

Realizamos un experimento adicional de modificación descartando los canales de color para verificar la validez del modelo en la identificación ratas en modelos en blanco y negro. Los resultados pueden verse en Figura 49.

Como resultado observable, se muestra que la eliminación de los canales de color parece tener poca o ninguna influencia. Cuando lo comparamos con los datos del entrenamiento estándar, la prueba mostró una pequeña mejora al usar imágenes en blanco y negro, pero esto en solo una diferencia del 0,09 %.

---

<sup>12</sup> La diferencia de precisión entre el modelo con mayor y menor precisión es solo del 0.15 %.

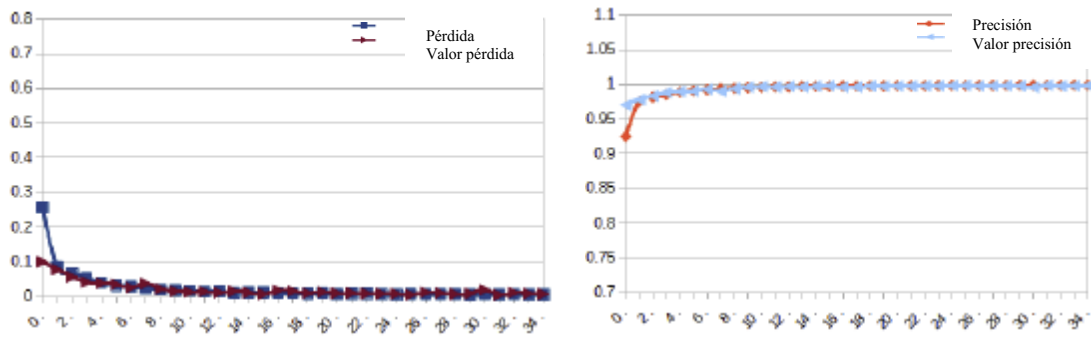


Figura 49. Pérdida y precisión por epoch para imágenes en blanco y negro.

## 6. Conclusiones

---

El software de seguimiento de animales es una parte fundamental de muchos estudios con animales. A pesar de ello existen limitaciones en este tipo de software como es el análisis de situaciones con múltiples animales. Debido a las dificultades en la preservación de la identidad de los mismos. En cualquier caso, en este trabajo se demuestra que, mediante el uso de las redes convolucionales de aprendizaje profundo, entrenadas con técnicas de transferencia del conocimiento, es posible abordar esta tarea.

Durante el desarrollo desempeñado se ha estudiado el estado del arte, la preparación de datos, la evaluación de los resultados, el diseño de múltiples pruebas de experimentación y el entrenamiento de los modelos. Todo ello ha sido necesario y cada tarea realizada es fundamental para el trabajo. En resumen, los pasos realizados en este trabajo han sido:

1. Comprender qué es lo que necesitamos realizar en el proyecto y qué relevancia tiene dentro del campo de las técnicas de seguimiento, enmarcado en el análisis del estado del arte.
2. Realizar un complejo proceso de visión artificial para crear conjuntos de datos.
3. Planificar y definir las pruebas a realizar La importancia de este proceso marca qué es lo que se va a estudiar durante la investigación. En este caso considero adecuadas las pruebas escogidas.
4. Entrenar los modelos y validarlos para la tarea de clasificación de ratas de laboratorio.

Los resultados obtenidos de éste proyecto pueden considerarse satisfactorios. En conclusión, las redes entrenadas lograron un buen desempeño en las pruebas finales de configuración, pudiendo llegar a identificar a las ratas de laboratorio con un porcentaje de precisión próximo al 95 %, aún con una muestra limitada de 281 imágenes. Según estos resultados me atrevo a concluir, y considerando las autolimitaciones de aprendizaje y los desafíos que planteaban los objetivos, que hemos logrado entrenar redes capaces de identificar. El objetivo de diseñar e integrar una alternativa útil y presumiblemente precisa ha sido alcanzado.

No obstante, hay que destacar que el marco teórico en este campo de conocimiento aún no ha concluido. Hay pruebas adicionales como aquellas que puedan incorporar diferentes sesiones u otras que consideren la influencia del ruido, la magnitud y número de imágenes requeridas que posibilitarían obtener más información y certezas.

Adicionalmente debería realizarse un ensayo donde entrenamiento y validación se realicen en videos diferentes para descartar la posibilidad de la existencia de variables ocultas que podrían contaminar los resultados de la identificación, ya que la red podría estar memorizando características específicas de los videos, en lugar de los animales.

# APÉNDICES

## 6.1. Resultados completos de las pruebas

### 6.1.1. Epochs

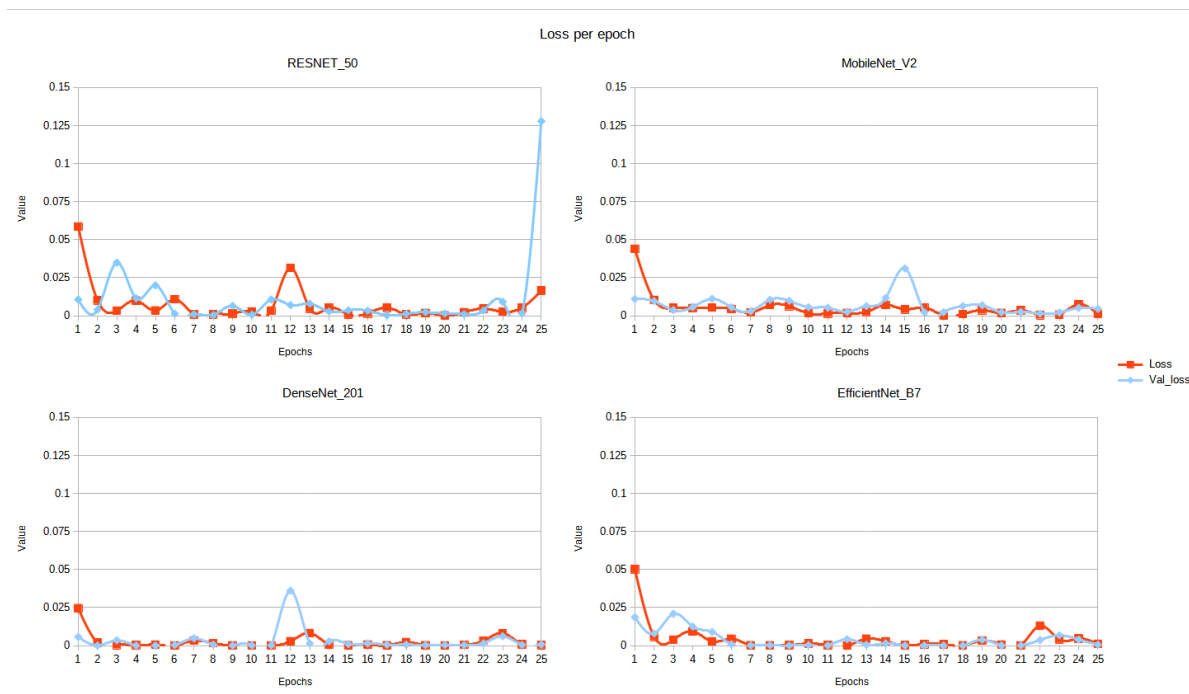


Figura 50 Curva de la función de pérdida para los ciclos empleados

### Pérdida y exactitud en 25 epochs de entrenamiento

Model	Loss	Accuracy	Val Loss	Val Accuracy
RESNET_50	0.016	0.99	0.12	0.94
MobileNet_V2	0.00	0.99	0.00	0.99
DenseNet_201	0.00	0.99	0.00	0.99
EfficientNet_B7	0.00	0.99	0.00	0.99

TABLA 10 PÉRDIDA Y EXACTITUD EN 25 EPOCHS DE ENTRENAMIENTO

### 6.1.2. Tamaño de imagen

Pérdida por tamaño de imagen por red



Model	32	64	128
RESNET_50	0.00	0.00	0.00
MobileNet_V2	0.00	0.00	0.00
DenseNet_201	0.00	0.00	
EfficientNet_B7	0.00	0.002	

TABLA 11 PÉRDIDA POR TAMAÑO DE IMAGEN POR RED

### Precisión por tamaño de imagen por red

Model	32	64	128
RESNET_50	0.99	0.99	0.99
MobileNet_V2	0.99	0.99	0.99
DenseNet_201	0.99	1	
EfficientNet_B7	0.99	0.99	

TABLA 12 PRECISIÓN POR TAMAÑO DE IMAGEN POR RED

### Gráficas

#### RESNET-50

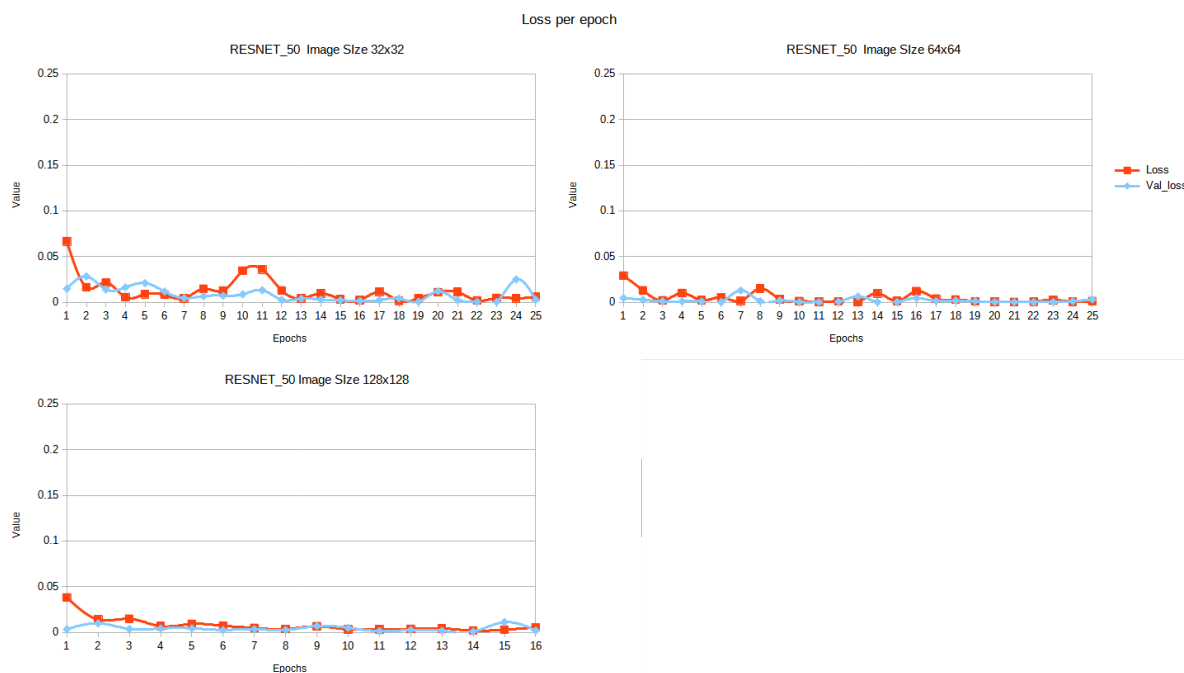
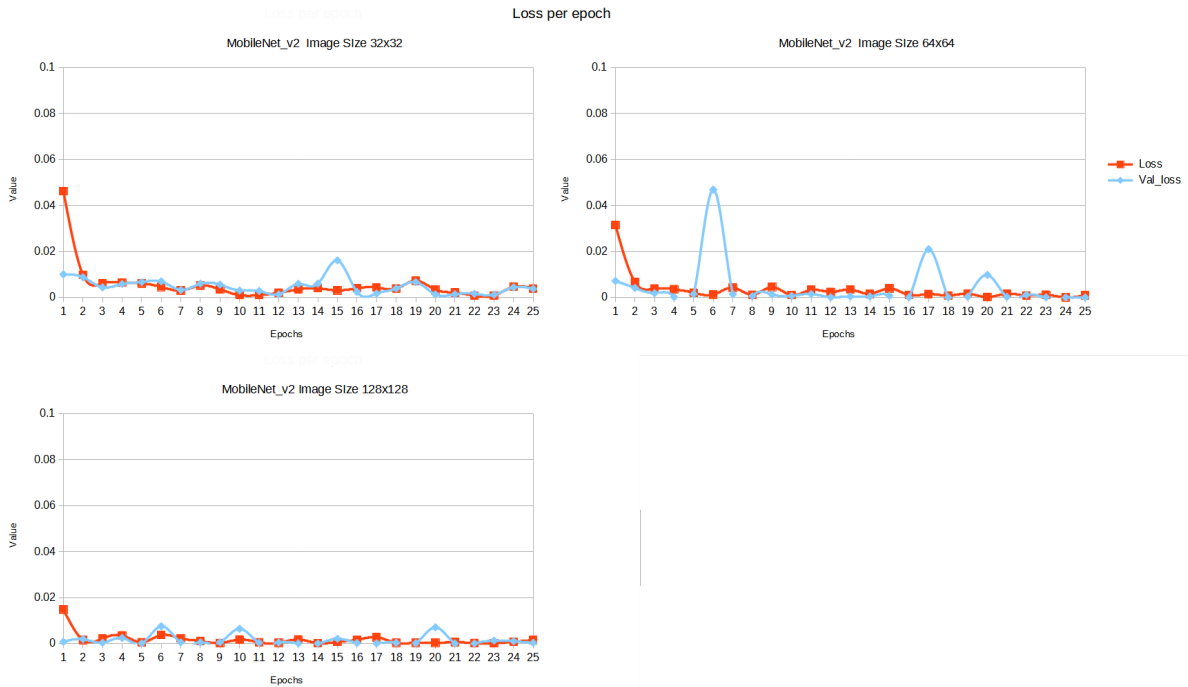


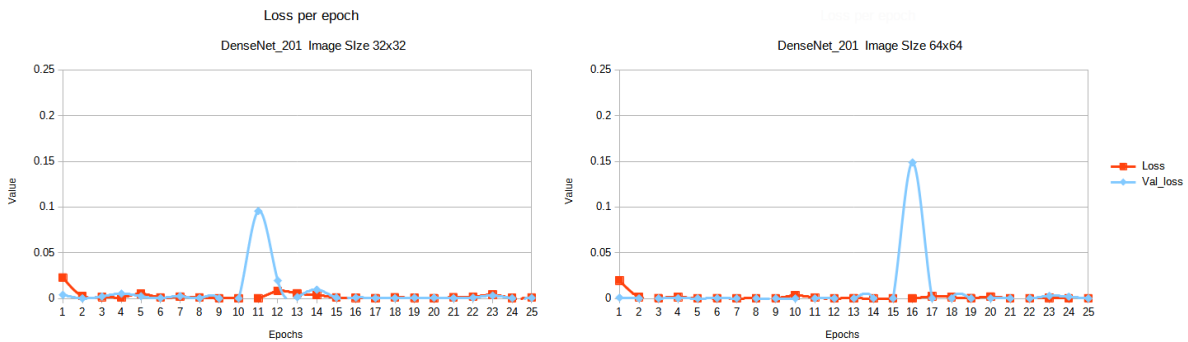
Figura 50 Pérdida para diferentes tamaños de imagen RESNET

#### MobileNet-V2



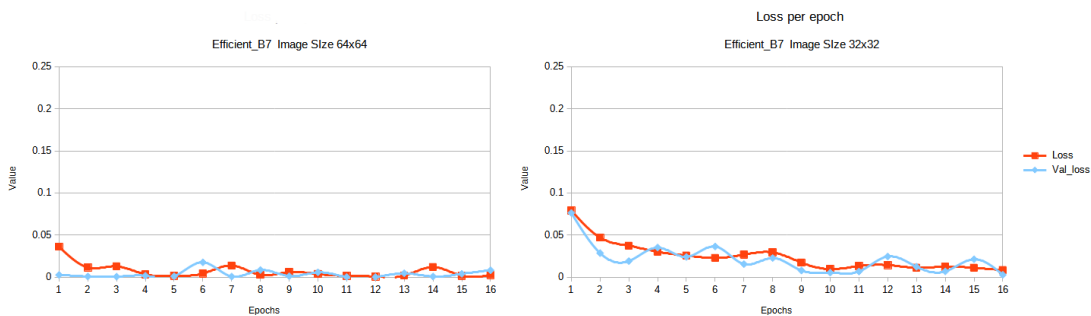
**Figura 51 Pérdida para diferentes tamaños de imagen MobileNet-V2**

**DenseNet-201**



**Figura 52 Pérdida para diferentes tamaños de imagen DenseNet**

**EfficientNet\_B7**



**Figura 53 Pérdida para diferentes tamaños de imagen EfficientNet**

Macro-promediadas  
Tamaño de imagen 32x32:

Tamaño de imagen 64x64:

	recall	precision	sensitivity	specificity	PPV	NPV	f1
RESNET	0.99	0.99	99.90	0.99	99.90	99.90	0.99
MobileNet	1	1	99.99	1	99.99	99.99	1
DenseNet	1	1	99.99	1	99.99	99.99	1
EfficientNet	0.99	0.99	99.9	0.99	99.96	99.96	0.99

TABLA 13 MÉTRICAS MACRO-PROMEDIADAS PARA TAMAÑO DE IMAGEN 64x64

Tamaño de imagen 128x128:

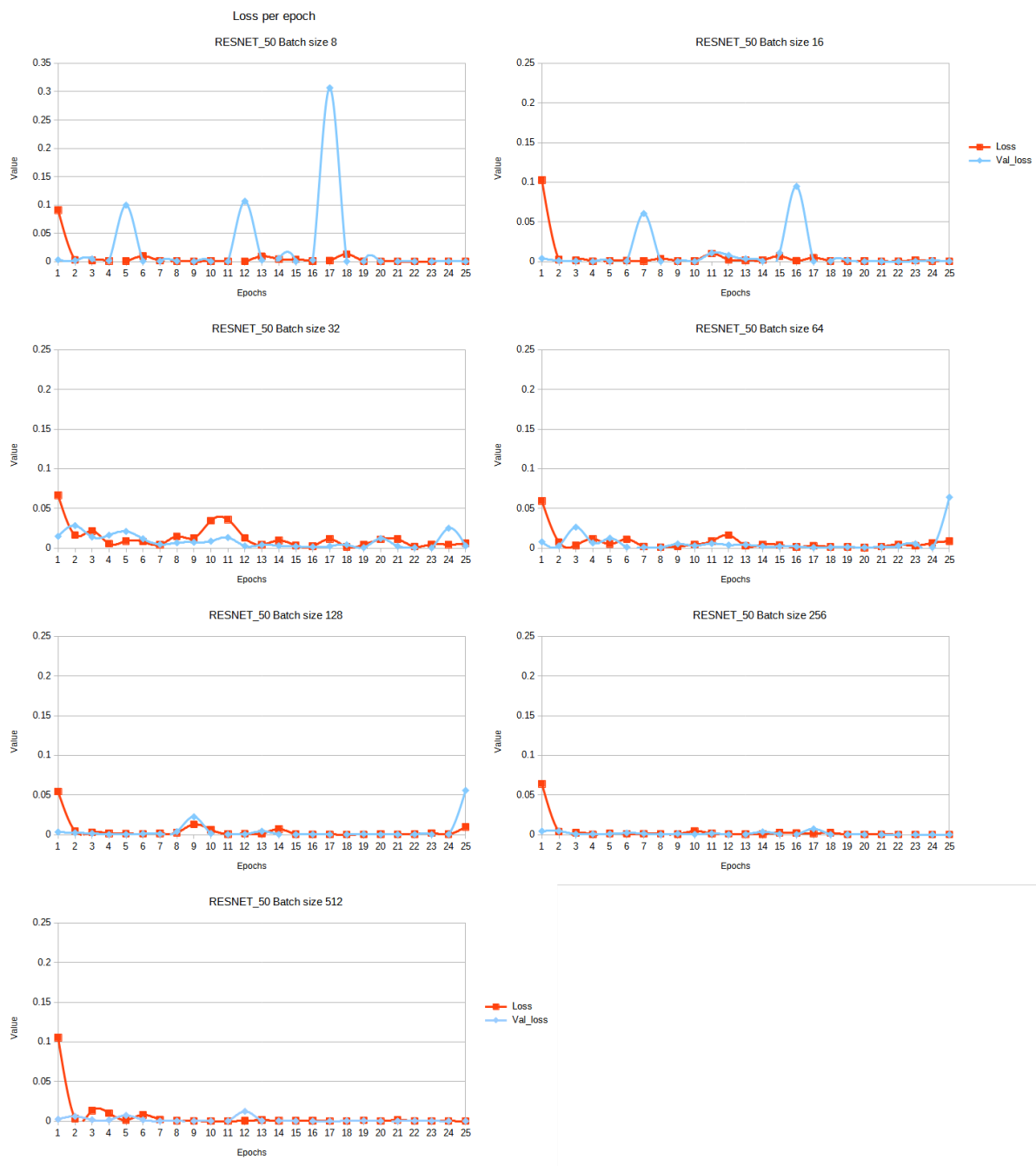
	recall	precision	sensitivity	specificity	PPV	NPV	f1
RESNET	0.99	0.99	99.97	0.99	99.97	99.97	0.99
MobileNet	0.999	0.99	99.99	0.999	99.99	99.99	0.99

TABLA 14 PÉRDIDA PARA DIFERENTES TAMAÑOS DE IMAGEN MOBILENET

### 6.1.3. Tamaño de muestra

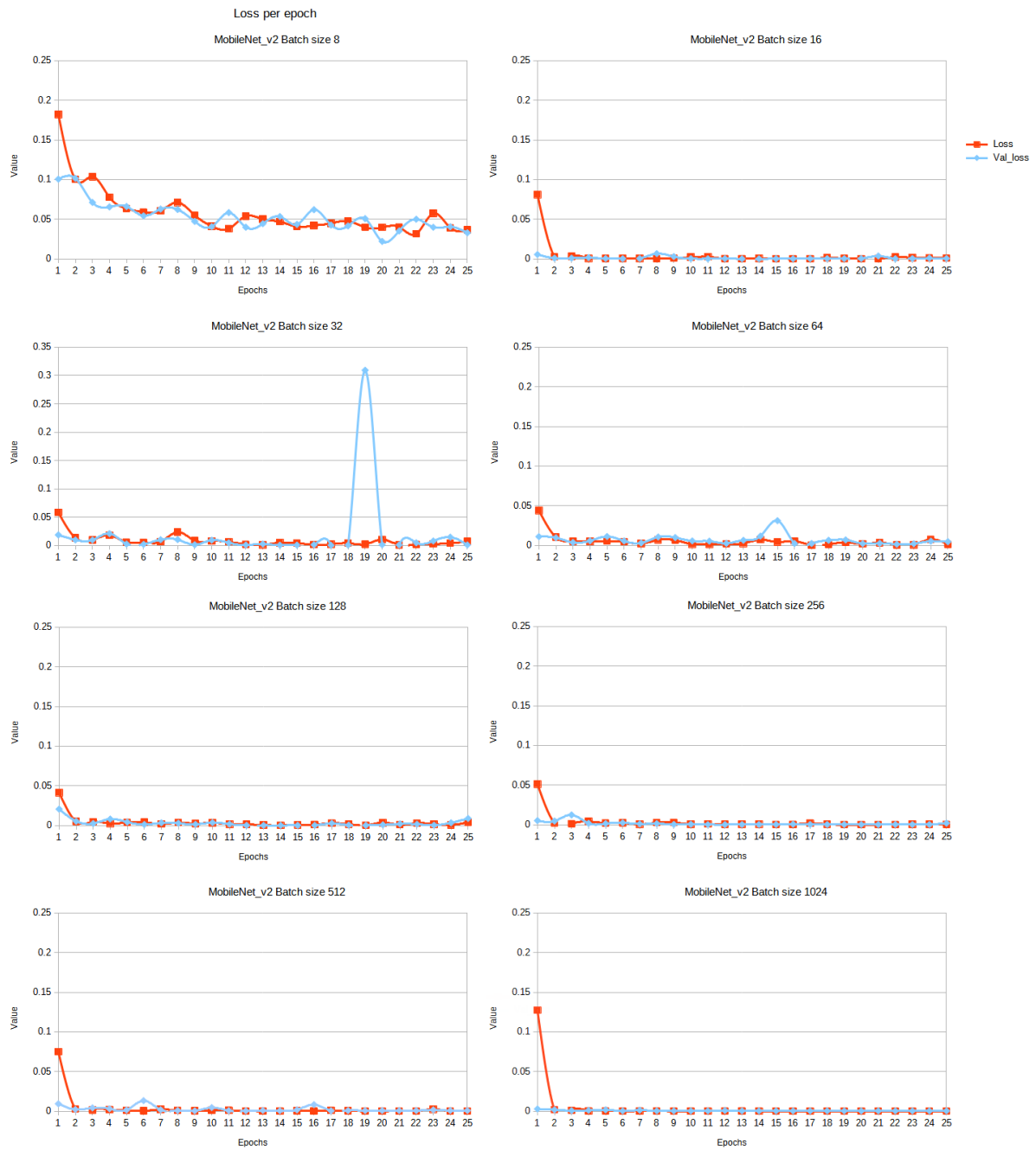
#### Graficas

Métricas macro-promediadas para tamaño de imagen 64x64  
RESNET-50



**Figura 53 Pérdida por tamaño de muestra RESNET**

# MobileNet-V2



**Figura 54 Pérdida por tamaño de muestra MobileNet**

# DenseNet\_201

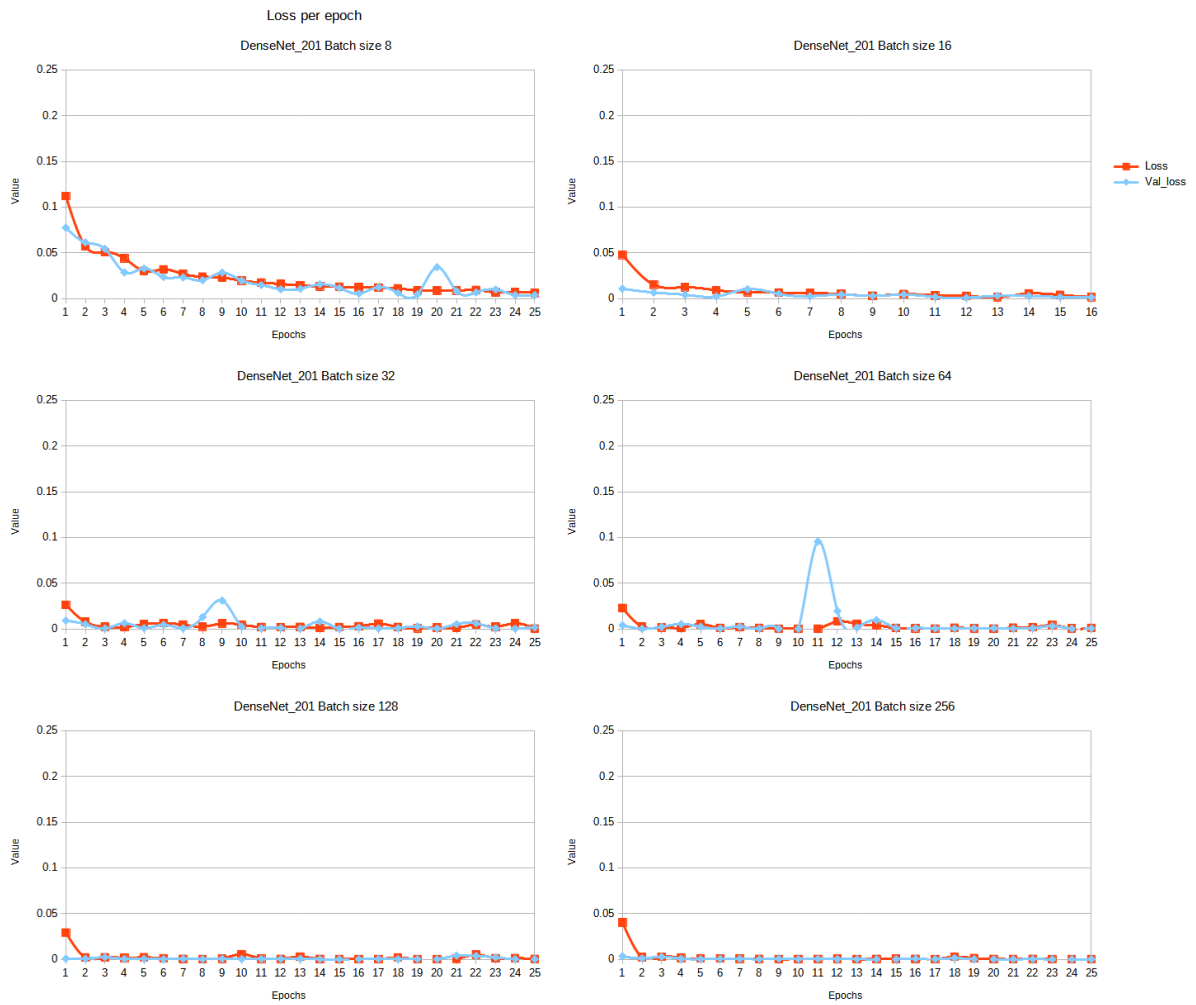


Figura 55 Pérdida por tamaño de muestra DenseNet

# EfficientNet\_B7

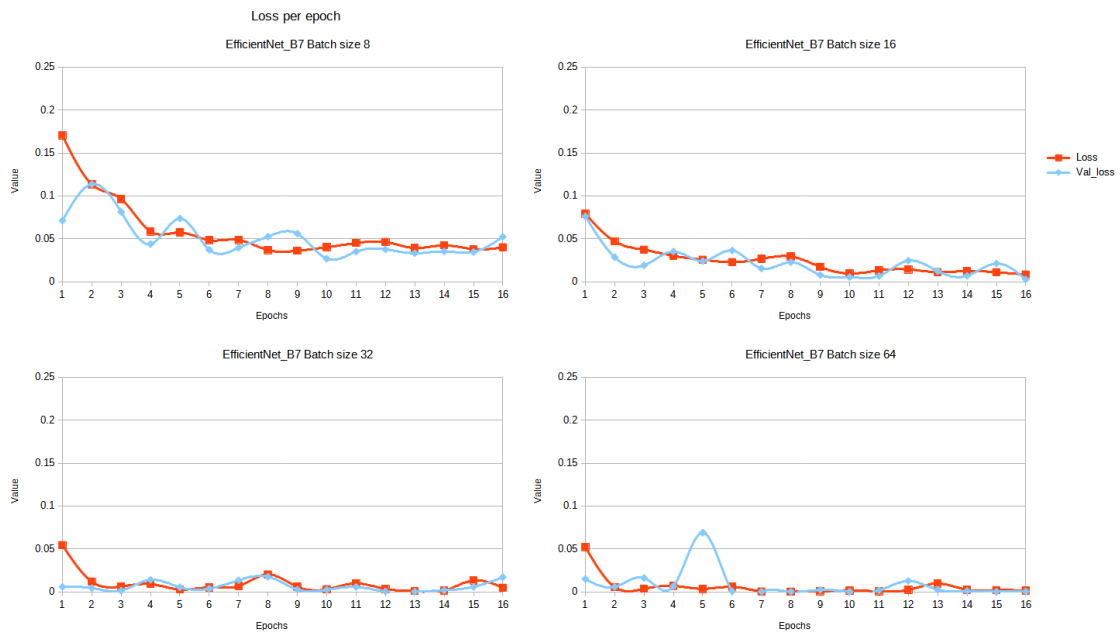


Figura 56 Pérdida por tamaño de muestra EfficientNet

## Métricas

RESNET\_50

### Macro-promediadas

tamaño de muestra	recall	precision	sensitivity	specificity	PPV	NPV	f1
8	0.99	0.99	99.72	0.99	99.72	99.72	0.99
16	0.99	0.99	99.75	0.99	99.75	99.75	0.99
32	0.99	0.99	99.95	0.99	99.95	99.95	0.99
64	0.97	0.97	97.22	0.97	97.90	97.22	0.97
128	0.97	0.97	97.42	0.97	97.65	97.42	0.97
256	0.99	0.99	99.88	0.99	99.88	99.88	0.99
512	0.99	0.99	99.74	0.99	99.74	99.74	0.99

TABLA 15 MÉTRICAS MACRO-PROMEDIADAS PARA TAMAÑOS DE MUESTRA RESNET

DenseNet\_201

### Macro-promediadas

tamaño de muestra	recall	precision	sensitivity	specificity	PPV	NPV	f1
8	0.99	0.99	99.93	0.99	99.93	99.93	0.99
16	0.99	0.99	99.77	0.99	99.78	99.77	0.99
32	0.99	0.99	99.94	0.99	99.94	99.94	0.99
64	0.99	0.99	99.96	0.99	99.96	99.96	0.99
128	1.00	1.00	99.97	1.00	99.97	99.97	1.00
256	0.99	0.99	99.93	0.99	99.93	99.93	0.99

TABLA 16 MÉTRICAS MACRO-PROMEDIADAS PARA TAMAÑOS DE MUESTRA DENSENET

EfficientNet\_B7

### Macro-promediadas

tamaño de muestra	recall	precision	sensitivity	specificity	PPV	NPV	f1
8	0.94	0.96	94.81	0.94	96.44	94.81	0.93
16	0.99	0.99	99.92	0.99	99.92	99.92	0.99
32	0.99	0.99	99.96	0.99	99.96	99.96	0.99
64	0.99	0.99	99.98	0.99	99.98	99.98	1.00

TABLA 17 MÉTRICAS MACRO-PROMEDIADAS PARA TAMAÑOS DE MUESTRA EFFICIENTNET



Model	Loss	Accuracy	Val Loss	Val Accuracy
RESNET_50	0.01	0.99	0.00	0.99
MobileNet_V2	0.01	0.99	0.01	0.99
DenseNet_201	0.00	0.99	0.00	0.99
<b>EfficientNet_B7</b>	<b>0.01</b>	<b>0.99</b>	<b>0.00</b>	<b>0.99</b>

TABLA 18 TABLA DE PÉRDIDA Y EXACTITUD A 16 EPOCHS PARA LA COMPARACIÓN

#### Tiempo de entrenamiento por epoch por red

Model	Time per epoch
RESNET_50	239.32 seconds
MobileNet_V2	55.08 seconds
DenseNet_201	463.09 seconds
EfficientNet_B7	753.58 seconds

TABLA 19 TIEMPO DE ENTRENAMIENTO POR EPOCH POR RED PARA LA COMPARACIÓN

#### Métricas macro-promediadas:

Model	True Positives	True Negatives	False Positives	False Negatives
RESNET_50	9977.89	149668.31	22.11	331.69
MobileNet_V2	9984.83	149772.47	15.17	227.53
DenseNet_201	9991.91	149878.69	8.09	121.31
EfficientNet_B7	9983.41	149751.19	16.59	248.81

TABLA 20 MEDIA DE VALORES PARA LA PRUEBA DE COMPARACIÓN

Model	recall	precision	sensitivity	specificity	PPV	NPV	f1
RESNET	0.96	0.99	96.81	0.99	99.78	99.98	0.98
MobileNet	0.97	0.99	97.80	1.00	99.84	99.98	0.98
DenseNet	0.98	0.99	98.80	1.00	99.91	99.99	0.99
<b>EfficientNet</b>	<b>0.97</b>	<b>0.99</b>	<b>97.57</b>	<b>1.00</b>	<b>99.83</b>	<b>99.98</b>	<b>0.98</b>

TABLA 21 MÉTRICAS MACRO-PROMEDIADAS PARA LA PRUEBA DE COMPARACIÓN

### 6.1.4. Prueba cambiando el número de individuos y de clases

Precisión por clase y numero de individuos

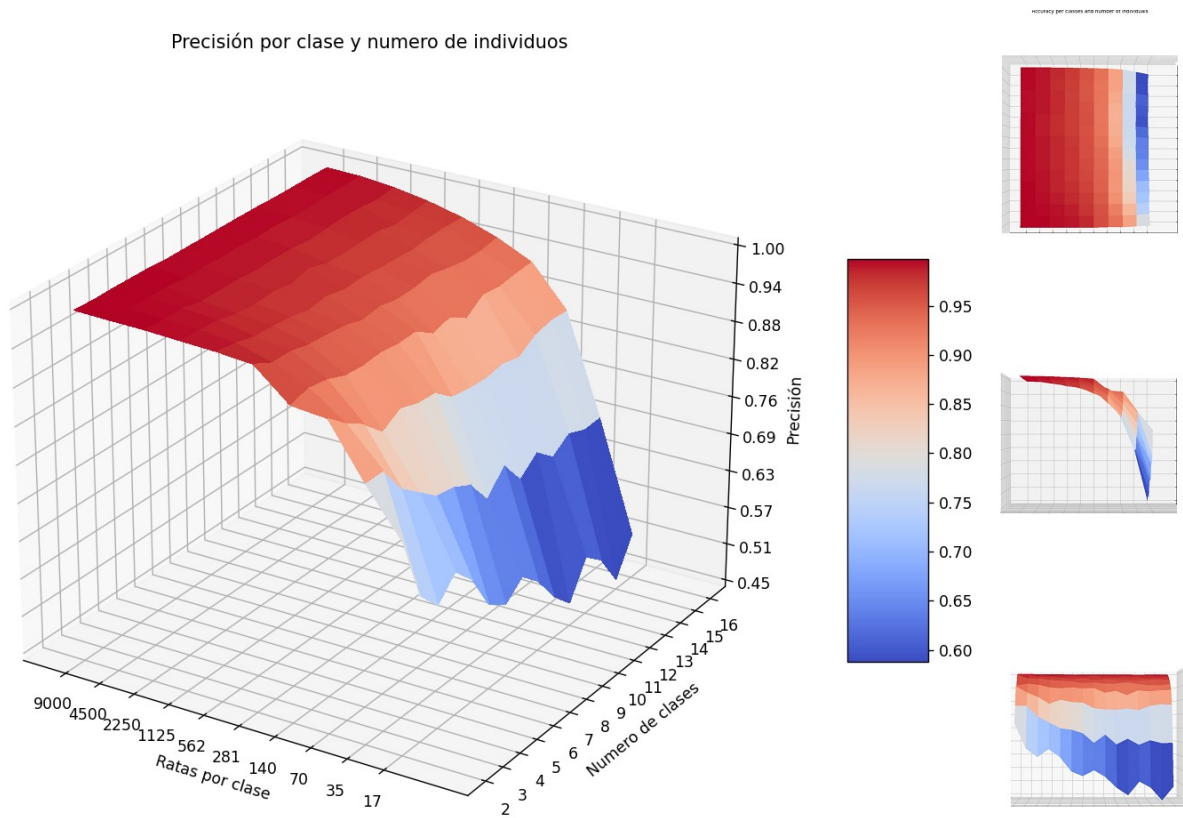


Figura 57 Gráfica 3D de precisión por número de clases y datos de entrenamiento

## Precisión de los modelos por número de clases y número de datos de entrenamiento

	9000	4500	2250	1125	562	281	140	70	35	17
2 clases	0.99	0.99	0.99	0.99	0.99	0.98	0.93	0.93	0.85	0.77
3 clases	0.99	0.99	0.99	0.99	0.98	0.97	0.94	0.90	0.85	0.67
4 clases	0.99	0.99	0.99	0.99	0.97	0.97	0.94	0.88	0.80	0.63
5 clases	0.99	0.99	0.98	0.98	0.97	0.96	0.92	0.85	0.77	0.66
6 clases	0.99	0.99	0.98	0.98	0.97	0.96	0.92	0.87	0.75	0.63
7 clases	0.99	0.99	0.98	0.98	0.96	0.95	0.93	0.86	0.74	0.57
8 clases	0.99	0.99	0.98	0.97	0.97	0.94	0.92	0.86	0.73	0.55
9 clases	0.99	0.99	0.98	0.98	0.96	0.95	0.92	0.84	0.69	0.58
10 clases	0.99	0.99	0.98	0.97	0.96	0.94	0.90	0.83	0.72	0.55
11 clases	0.99	0.99	0.98	0.97	0.96	0.94	0.91	0.83	0.68	0.50
12 clases	0.99	0.99	0.98	0.97	0.96	0.95	0.89	0.83	0.71	0.47
13 clases	0.99	0.99	0.98	0.97	0.96	0.94	0.91	0.83	0.68	0.53
14 clases	0.99	0.99	0.98	0.97	0.96	0.94	0.91	0.84	0.70	0.50
15 clases	0.99	0.99	0.98	0.97	0.96	0.94	0.91	0.85	0.69	0.45
16 clases	0.99	0.99	0.98	0.98	0.96	0.94	0.91	0.85	0.69	0.51

TABLA 22 PRECISIÓN POR CLASE Y NÚMERO DE DATOS

### 6.1.5. Pruebas con imágenes en blanco y negro

Test Results for black and white images

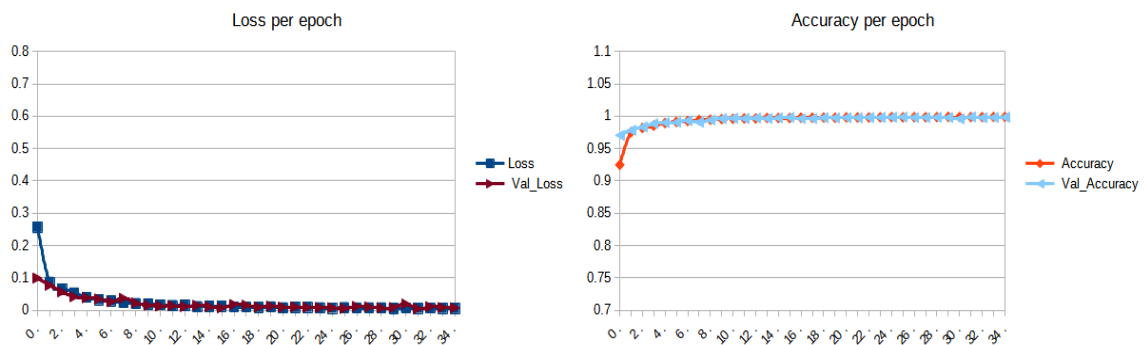


Figura 58 Pérdida y precisión para el caso blanco y negro

## Métricas

Model	Loss	Accuracy	Val Loss	Val Accuracy
Imágenes a color	0.01	0.99	0.01	0.99
Blanco y negro	0.01	0.99	0.00	0.99

TABLA 23 PÉRDIDA Y PRECISIÓN PARA LOS RESULTADOS DE COLOR Y BLANCO Y NEGRO

Model	recall	precision	sensitivity	specificity	PPV	NPV	f1
Blanco y negro	0.96	0.99	96.33	0.99	99.74	99.98	0.98

TABLA 24 MÉTRICAS PARA LA PRUEBA DE BLANCO Y NEGRO

# Relación de acrónimos

**CUDA** [Compute Unified Device Architecture](#)

**CNN** [Convolutional Neural Network](#)

**GPU** [Graphical Processing Unit](#)

**MPEG4** [Moving Picture Experts Group](#)

**RESNET50** [Residual Network](#)

**ResNet** [Residual neural network](#)

**RNN** [Recurrent Neural Network](#)

**K-NN** [K- Nearest neighbor](#)

# Glosario

**Algoritmo** Conjunto ordenado y finito de operaciones que permiten hallar la solución de un problema[52].

**Algoritmos de seguimiento** Conjunto de algoritmos o procesos que se emplean para determinar la posición y comportamiento de objetos o individuos.

**Aprendizaje Máquina / Aprendizaje Automático** Aprendizaje automático, en Inteligencia Artificial, disciplina que se ocupa de la implementación de software informático que puede aprender y replicar de forma autónoma.

**Aprendizaje Profundo o *Deep learning*** Son un subconjunto de Aprendizaje Máquina que constituye una red neuronal con tres o más capas. Estas redes neuronales intentan emular el comportamiento del cerebro humano —aunque lejos de igualar su capacidad— pero le permiten "aprender y gestionar" a partir de grandes cantidades de datos.”[41]

**Bias de activación** Valor constante que permite cambiar el umbral de activación de neurona.

**Capacidad de generalización** Idoneidad de las redes neuronales para entender las relaciones entre los datos.

**Codificar** Verbo: acto de escribir código 40.

**Código** En la programación de computadoras, el código de computadora se refiere al conjunto de instrucciones, o un sistema de reglas, escritas en un lenguaje de programación discreto.

**Compute Unified Device Architecture** Plataforma de computación paralela e interfaz de programación de aplicaciones (API) que permite que el software use ciertos tipos de unidades de procesamiento de gráficos (GPU) para el procesamiento de propósito general.

**Cudnn** La biblioteca NVIDIA CUDA® Deep Neural Network (cuDNN) es una biblioteca de primitivas aceleradas por GPU para redes neuronales

profundas. cuDNN proporciona implementaciones altamente ajustadas para rutinas estándar, como capas de convolución hacia adelante y hacia atrás, agrupación, normalización y activación.

**Elemento estructurante** El elemento estructurante consiste en un patrón especificado como las coordenadas de un número de puntos discretos relativos a algún origen.

**Filtro de entropía** Filtro que acentúa los bordes iluminando los píxeles que tienen vecinos diferentes.

**Graphical Processing Unit** Componente de un ordenador dedicado a realizar operaciones relacionadas con gráficos y operaciones de coma flotante.

**Ground truth** Anglicismo. Conjunto de hechos verdaderos.

**Imágenes binarias** Las imágenes binarias son imágenes cuyos píxeles tienen solo dos valores de intensidad posibles.

**Imágenes en escala de gris** La escala de grises es la representación de una imagen en la que cada pixel se dibuja usando un valor numérico individual que representa su luminancia, en una escala que se extiende entre blanco y negro[53]

**Keras** Paquete de código relacionado con el Aprendizaje Máquina[54].

**Kernel** “Operador matemático que produce una nueva función  $f(y)$  integrando el producto de una función existente  $F(x)$  y la llamada función kernel  $K(x, y)$ ”

**K nearest neighbour** El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un clasificador de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual[55].

**Macro-promediado** Función que agrega el valor de una métrica para todas las clases. Por ejemplo: “La puntuación F1 macro promediada (o puntuación F1 macro) se calcula utilizando la media aritmética (también conocida como media no ponderada) de todas las puntuaciones F1 por clase [56].

**Micro-promediado** Por contra el micro promediado calcula una puntuación F1 media global contando las sumas de los verdaderos positivos (TP), los Falsos Negativos (FN) y los Falsos Positivos (FP) [56].

**Redes neuronales** “Red neuronal, un programa informático que funciona de una manera inspirada en la red neuronal natural del cerebro. El objetivo de estas redes neuronales artificiales es realizar funciones cognitivas como la resolución de problemas y el aprendizaje automático [57].

**Recurrent Neural Network** redes que se caracterizan por estar conectadas en ciclos bidireccionales.

**Residual Network** Red neuronal perteneciente a la familia de redes de neuronas residuales. Éstas se componen de un total de cincuenta capas siendo cuarenta y ocho capas convolucionales, una capa MaxPool y una capa mediadora.

**Ruido** El ruido de la imagen es una variación aleatoria de la información de brillo o color en las imágenes capturadas.

**Software** Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora[58].

**Técnicas de suavizado** Serie de técnicas y procesos que se utilizan para eliminar el ruido en imágenes.

**Threshold** Término anglosajón para Umbral. Es el valor a partir del cual se realiza un corte.

**Píxel** Elementos de visualización pequeños, cuadrados y de un solo color que componen una imagen [59].

**Python** Python es un lenguaje de alto nivel de programación interpretado.

**Operación morfológica** Las operaciones morfológicas son un conjunto amplio de operaciones que sirven para procesar imágenes basadas en formas geométricas.

**Validación cruzada** La validación cruzada, también conocida como cross validation, es una técnica que se usa en el aprendizaje automático (machine learning) para evaluar la variabilidad de los datos y la confiabilidad de cualquier modelo que se esté utilizando [60].



**Visión artificial** Campo de la inteligencia artificial en el que los programas intentan identificar objetos representados en imágenes digitalizadas proporcionadas por cámaras [\[61\]](#).

# Bibliografía

- [1] C. Ari, D. P. D'Agostino, D. M. Diamond, M. Kindy, C. Park, and Z. Kovács, 'Elevated Plus Maze Test Combined with Video Tracking Software to Investigate the Anxiolytic Effect of Exogenous Ketogenic Supplements', *JoVE J. Vis. Exp.*, no. 143, p. e58396, Jan. 2019, doi: 10.3791/58396.
- [2] M. Pometlová, K. Nohejlová-Deykun, and R. Šlamberová, 'Anxiogenic Effect of Low-dose Methamphetamine in the Test of Elevated Plus-maze', *Prague Med. Rep.*, vol. 113, no. 3, pp. 223–230, 2012, doi: 10.14712/23362936.2015.20.
- [3] R. J. Rodgers and A. Dalvi, 'Anxiety, defence and the elevated plus-maze', *Neurosci. Biobehav. Rev.*, vol. 21, no. 6, pp. 801–810, Nov. 1997, doi: 10.1016/S0149-7634(96)00058-9.
- [4] 'The feeding behaviour of elephant (*Loxodonta africana*) in the Sengwa Area Rhodesia'. [https://journals.co.za/doi/epdf/10.10520/AJA03794369\\_3337](https://journals.co.za/doi/epdf/10.10520/AJA03794369_3337) (accessed Jun. 08, 2023).
- [5] B. Sullivan, M. Iliff, C. Wood, D. Fink, and S. Kelling, 'eBird---Using citizen-science data to help solve real-world conservation challenges (Invited)', *AGU Fall Meet. Abstr.*, Dec. 2010.
- [6] R. G. Farmer, M. L. Leonard, and A. G. Horn, 'Observer Effects and Avian-Call-Count Survey Quality: Rare-Species Biases and Overconfidence', Jan. 2012, doi: 10.1525/auk.2012.11129.
- [7] P. F. Hadaway, B. K. Alexander, R. B. Coombs, and B. Beyerstein, 'The effect of housing and gender on preference for morphine-sucrose solutions in rats', *Psychopharmacology (Berl.)*, vol. 66, no. 1, pp. 87–91, Nov. 1979, doi: 10.1007/BF00431995.
- [8] J. R. Alberts, 'Huddling by rat pups: Multisensory control of contact behavior', *J. Comp. Physiol. Psychol.*, vol. 92, no. 2, pp. 220–230, 1978, doi: 10.1037/h0077458.
- [9] R. J. Blanchard, K. J. Flannelly, and D. C. Blanchard, 'Life-span studies of dominance and aggression in established colonies of laboratory rats', *Physiol. Behav.*, vol. 43, no. 1, pp. 1–7, Jan. 1988, doi: 10.1016/0031-9384(88)90089-3.
- [10] D. B. Dusenbery, 'Using a microcomputer and video camera to simultaneously track 25 animals', *Comput. Biol. Med.*, vol. 15, no. 4, pp. 169–175, Jan. 1985, doi: 10.1016/0010-4825(85)90058-7.
- [11] I. Powers, 'MC6809 microprocessor', *Microprocessors*, vol. 2, no. 3, p. 162, Jul. 1978, doi: 10.1016/0308-5953(78)90010-7.
- [12] E. Bakchine-Huber, F. Marion-Poll, M. H. Pham-Delegue, C. Masson, and F. Soussaline, 'Real-time detection and analysis of the exploratory behavior of small animals: An application to the study of the olfactory behavior of honeybees in a four-choice device', *Naturwissenschaften*, vol. 79, no. 1, pp. 39–42, Jan. 1992, doi: 10.1007/BF01132282.

- [13] [Y. Günzel, F. B. Oberhauser, and E. Couzin-Fuchs, 'Information integration for decision-making in desert locusts', \*iScience\*, vol. 26, no. 4, p. 106388, Apr. 2023, doi: 10.1016/j.isci.2023.106388.](#)
- [14] [N. M. Lind, M. Vinther, R. P. Hemmingsen, and A. K. Hansen, 'Validation of a digital video tracking system for recording pig locomotor behaviour', \*J. Neurosci. Methods\*, vol. 143, no. 2, pp. 123–132, Apr. 2005, doi: 10.1016/j.jneumeth.2004.09.019.](#)
- [15] [B. Q. Geuther \*et al.\*, 'Robust mouse tracking in complex environments using neural networks', \*Commun. Biol.\*, vol. 2, no. 1, Art. no. 1, Mar. 2019, doi: 10.1038/s42003-019-0362-1.](#)
- [16] [T. D. Pereira \*et al.\*, 'Fast animal pose estimation using deep neural networks', \*Nat. Methods\*, vol. 16, no. 1, Art. no. 1, Jan. 2019, doi: 10.1038/s41592-018-0234-5.](#)
- [17] [K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, 'Activity Forecasting', in \*Computer Vision – ECCV 2012\*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., in \*Lecture Notes in Computer Science\*. Berlin, Heidelberg: Springer, 2012, pp. 201–214. doi: 10.1007/978-3-642-33765-9\\_15.](#)
- [18] [P. Bergmann, T. Meinhardt, and L. Leal-Taixé, 'Tracking Without Bells and Whistles', in \*2019 IEEE/CVF International Conference on Computer Vision \(ICCV\)\*, Oct. 2019, pp. 941–951. doi: 10.1109/ICCV.2019.00103.](#)
- [19] ['ABCTracker'. https://www.abctracker.org/ \(accessed Jul. 17, 2023\).](https://www.abctracker.org/)
- [20] [E. Itskovits, A. Levine, E. Cohen, and A. Zaslaver, 'A multi-animal tracker for studying complex behaviors', \*BMC Biol.\*, vol. 15, no. 1, p. 29, Apr. 2017, doi: 10.1186/s12915-017-0363-9.](#)
- [21] [A. M. T. Harmer and D. B. Thomas, 'pathtrackr: An r package for video tracking and analysing animal movement', \*Methods Ecol. Evol.\*, vol. 10, no. 8, Art. no. 8, 2019.](#)
- [22] [Q. He, Q. Zhao, N. Liu, P. Chen, Z. Zhang, and R. Hou, 'Distinguishing Individual Red Pandas from Their Faces', in \*Pattern Recognition and Computer Vision\*, Z. Lin, L. Wang, J. Yang, G. Shi, T. Tan, N. Zheng, X. Chen, and Y. Zhang, Eds., in \*Lecture Notes in Computer Science\*. Cham: Springer International Publishing, 2019, pp. 714–724. doi: 10.1007/978-3-030-31723-2\\_61.](#)
- [23] [A. Freytag, E. Rodner, M. Simon, A. Loos, H. S. Köhl, and J. Denzler, 'Chimpanzee Faces in the Wild: Log-Euclidean CNNs for Predicting Identities and Attributes of Primates', in \*Pattern Recognition\*, B. Rosenhahn and B. Andres, Eds., in \*Lecture Notes in Computer Science\*. Cham: Springer International Publishing, 2016, pp. 51–63. doi: 10.1007/978-3-319-45886-1\\_5.](#)
- [24] [C.-A. Brust \*et al.\*, 'Towards Automated Visual Monitoring of Individual Gorillas in the Wild', in \*2017 IEEE International Conference on Computer Vision Workshops \(ICCVW\)\*, Oct. 2017, pp. 2820–2830. doi: 10.1109/ICCVW.2017.333.](#)
- [25] ['\[1812.04418\] Towards Automatic Identification of Elephants in the Wild'. https://arxiv.org/abs/1812.04418 \(accessed Aug. 10, 2023\).](#)
- [26] [L. Bergamini \*et al.\*, 'Multi-views Embedding for Cattle Re-identification', in](#)

- [2018 14th International Conference on Signal-Image Technology & Internet-Based Systems \(SITIS\), Nov. 2018, pp. 184–191. doi: 10.1109/SITIS.2018.00036.](#)
- [27] [Z. Li, C. Ge, S. Shen, and X. Li, ‘Cow Individual Identification Based on Convolutional Neural Network’, in \*Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence\*, in ACAI ’18, New York, NY, USA: Association for Computing Machinery, Diciembre 2018, pp. 1–5. doi: 10.1145/3302425.3302460.](#)
- [28] [C. N. Phyo, T. T. Zin, H. Hama, and I. Kobayashi, ‘A Hybrid Rolling Skew Histogram-Neural Network Approach to Dairy Cow Identification System’, in \*2018 International Conference on Image and Vision Computing New Zealand \(IVCNZ\)\*, Nov. 2018, pp. 1–5. doi: 10.1109/IVCNZ.2018.8634739.](#)
- [29] [S. Bouma, M. D. M. Pawley, K. Hupman, and A. Gilman, ‘Individual Common Dolphin Identification Via Metric Embedding Learning’, in \*2018 International Conference on Image and Vision Computing New Zealand \(IVCNZ\)\*, Nov. 2018, pp. 1–6. doi: 10.1109/IVCNZ.2018.8634778.](#)
- [30] [D. A. Konovalov, S. Hillcoat, G. Williams, R. A. Birtles, N. Gardiner, and M. I. Curnock, ‘Individual Minke Whale Recognition Using Deep Learning Convolutional Neural Networks’, \*J. Geosci. Environ. Prot.\*, vol. 6, no. 5, Art. no. 5, May 2018, doi: 10.4236/gep.2018.65003.](#)
- [31] [‘¿Qué es la visión artificial? | IBM’. <https://www.ibm.com/topics/computer-vision> \(accessed Jan. 04, 2023\).](#)
- [32] [V. Panadeiro, A. Rodriguez, J. Henry, D. Wlodkowic, and M. Andersson, ‘A review of 28 free animal-tracking software applications: current features and limitations’, \*Lab Anim.\*, vol. 50, no. 9, Art. no. 9, Sep. 2021, doi: 10.1038/s41684-021-00811-1.](#)
- [33] [Z. Zhang, ‘A flexible new technique for camera calibration’, \*IEEE Trans. Pattern Anal. Mach. Intell.\*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000, doi: 10.1109/34.888718.](#)
- [34] [A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, and G. G. de Polavieja, ‘idTracker: tracking individuals in a group by automatic identification of unmarked animals’, \*Nat. Methods\*, vol. 11, no. 7, Art. no. 7, Jul. 2014, doi: 10.1038/nmeth.2994.](#)
- [35] [A. Rodriguez, H. Zhang, J. Klaminder, T. Brodin, P. L. Andersson, and M. Andersson, ‘ToxTrac: A fast and robust software for tracking organisms’, \*Methods Ecol. Evol.\*, vol. 9, no. 3, pp. 460–464, 2018, doi: 10.1111/2041-210X.12874.](#)
- [36] [A. Kumar, ‘Most Common Types of Machine Learning Problems’, \*Data Analytics\*, Jan. 14, 2021. <https://vitalflux.com/most-common-types-machine-learning-problems/> \(accessed Dec. 05, 2022\).](#)
- [37] [‘What is Unsupervised Learning?’, Mar. 31, 2022. <https://www.ibm.com/cloud/learn/unsupervised-learning> \(accessed Nov. 15, 2022\).](#)
- [38] [‘What is Supervised Learning?’, Jun. 30, 2021. <https://www.ibm.com/cloud/learn/supervised-learning> \(accessed Nov. 15, 2022\).](#)
- [39] [T. Mummert, D. Subramanian, L. Vu, N. P. P. 14 de septiembre de 2022 T. save](#)

- [your progress, and you must be logged in to I. D. L. in now, ‘What is reinforcement learning?’, \*IBM Developer\*, Sep. 15, 2022. <https://developer.ibm.com/what-is-automated-ai-for-decision-making> \(accessed Nov. 22, 2022\).](#)
- [40] [‘What is a neuron?’, Nov. 22, 2016. <https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron> \(accessed Jan. 20, 2023\).](#)
- [41] [‘What is Deep Learning? | IBM’. <https://www.ibm.com/topics/deep-learning> \(accessed Jan. 10, 2023\).](#)
- [42] [‘3D Visualization of a Convolutional Neural Network’. \[https://adamharley.com/nn\\\_vis/cnn/3d.html\]\(https://adamharley.com/nn\_vis/cnn/3d.html\) \(accessed May 02, 2023\).](#)
- [43] [T. rédac, ‘¿Qué es el Transfer Learning?’, \*Formation Data Science | DataScientest.com\*, Jan. 06, 2022. <https://datascientest.com/es/que-es-el-transfer-learning> \(accessed Jun. 11, 2023\).](#)
- [44] [M. Sokolova and G. Lapalme, ‘A systematic analysis of performance measures for classification tasks’, \*Inf. Process. Manag.\*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.](#)
- [45] [B. Boehm and W. J. Hansen, ‘Spiral Development: Experience, Principles, and Refinements’, Defense Technical Information Center, Fort Belvoir, VA, Jul. 2000, doi: 10.21236/ADA382590.](#)
- [46] [‘cuda-msvc-compatibility’. <https://quasar.ugent.be/files/doc/cuda-msvc-compatibility.html> \(accessed Jan. 02, 2023\).](#)
- [47] [‘Descargas anteriores de Visual Studio: 2019, 2017, 2015 y versiones anteriores’, \*Visual Studio\*. <https://visualstudio.microsoft.com/es/vs/older-downloads/> \(accessed Jan. 02, 2023\).](#)
- [48] [Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, ‘CIFAR-10: KNN-Based Ensemble of Classifiers’, in \*2016 International Conference on Computational Science and Computational Intelligence \(CSCI\)\*, Dec. 2016, pp. 1192–1195. doi: 10.1109/CSCI.2016.0225.](#)
- [49] [R. C. Çalik and M. F. Demirci, ‘Cifar-10 Image Classification with Convolutional Neural Networks for Embedded Systems’, in \*2018 IEEE/ACS 15th International Conference on Computer Systems and Applications \(AICCSA\)\*, Oct. 2018, pp. 1–2. doi: 10.1109/AICCSA.2018.8612873.](#)
- [50] [‘Preservación da identidade animal particular en ensaios de comportamento social’. <https://www.udc.es/tfe/traballo/?codigo=23288> \(accessed May 15, 2023\).](#)
- [51] [J. Liu, F. Guo, H. Gao, M. Li, Y. Zhang, and H. Zhou, ‘Defect detection of injection molding products on small datasets using transfer learning’, \*J. Manuf. Process.\*, vol. 70, pp. 400–413, Oct. 2021, doi: 10.1016/j.jmapro.2021.08.034.](#)
- [52] [R.- ASALE and RAE, ‘algoritmo | Diccionario de la lengua española’, \*«Diccionario de la lengua española» - Edición del Tricentenario\*. <https://dle.rae.es/algoritmo> \(accessed Jun. 11, 2023\).](#)
- [53] [‘Escala de grises \(grayscale\) - Diccionario de fotografía y diseño’. <https://www.fotonostra.com/glosario/escalagrisesgrayscale.htm> \(accessed Jun. 11, 2023\).](#)

- [54] [‘Keras: Deep Learning for humans’](https://keras.io/). <https://keras.io/> (accessed Jun. 11, 2023).
- [55] [‘¿Qué es el algoritmo de k vecinos más cercanos? | IBM’](https://www.ibm.com/es-es/topics/knn). <https://www.ibm.com/es-es/topics/knn> (accessed Jun. 29, 2023).
- [56] K. Leung, ‘Micro, Macro & Weighted Averages of F1 Score, Clearly Explained’, *Medium*, Sep. 13, 2022. <https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f> (accessed Jun. 12, 2023).
- [57] [‘Neural network | Computing & Machine Learning | Britannica’](https://www.britannica.com/technology/neural-network), May 30, 2023. <https://www.britannica.com/technology/neural-network> (accessed Jun. 11, 2023).
- [58] R.- ASALE and RAE, ‘software | Diccionario de la lengua española’, *«Diccionario de la lengua española» - Edición del Tricentenario*. <https://dle.rae.es/software> (accessed Jun. 11, 2023).
- [59] [‘pixel | Search Online Etymology Dictionary’](https://www.etymonline.com/search?q=pixel). <https://www.etymonline.com/search?q=pixel> (accessed Jun. 11, 2023).
- [60] [‘Cross validation: qué es y su relación con machine learning’](https://blog.maestriasydiplomados.tec.mx/cross-validation-que-es-y-su-relacion-con-machine-learning). <https://blog.maestriasydiplomados.tec.mx/cross-validation-que-es-y-su-relacion-con-machine-learning> (accessed Jun. 11, 2023).
- [61] [‘Computer vision | Definition, Examples, Applications, & Facts | Britannica’](https://www.britannica.com/technology/computer-vision). <https://www.britannica.com/technology/computer-vision> (accessed Jun. 11, 2023).