PROYECTO FINAL

Redes Neuronales Convolucionales

GRUPO 10

21.12

ÍNDICE

ÍNDICE

ÍNDICE

ÍNDICE

ÍNDICE

ÍNDICE

ÍNDICE

¿Qué son las CNN?

¿Qué son las CNN?

Las capas convolucionales constituyen una herramienta invaluable cuando se trata de analizar datos que poseen estructuras espaciales.

¿Qué son las CNN?

Las capas convolucionales constituyen una herramienta invaluable cuando se trata de analizar datos que poseen estructuras espaciales.

Inspiradas en la capacidad del cerebro humano para reconocer patrones visuales, estas capas aplican filtros a regiones específicas de la entrada, permitiendo la detección de características locales.

Funcionamiento de la red

Funcionamiento de la red

Capas Lineales

Funcionamiento de la red

Capas Lineales

Cada neurona en una capa lineal está conectada a todas las neuronas de la capa anterior, aplicando una transformación lineal a los datos de entrada.

Funcionamiento de la red

Capas Lineales

Cada neurona en una capa lineal está conectada a todas las neuronas de la capa anterior, aplicando una transformación lineal a los datos de entrada.

$$y = Wx + b$$

Donde:

- y es la salida.
- W es la matriz de pesos.
- x es el vector de entrada.
- b es el sesgo.

Funcionamiento de la red

Capas Lineales

Cada neurona en una capa lineal está conectada a todas las neuronas de la capa anterior, aplicando una transformación lineal a los datos de entrada.

$$y = Wx + b$$

Donde:

- y es la salida.
- W es la matriz de pesos.
- x es el vector de entrada.
- b es el sesgo.

Capas convolucionales

Funcionamiento de la red

Capas Lineales

Cada neurona en una capa lineal está conectada a todas las neuronas de la capa anterior, aplicando una transformación lineal a los datos de entrada.

$$y = Wx + b$$

Donde:

- y es la salida.
- W es la matriz de pesos.
- x es el vector de entrada.
- b es el sesgo.

Capas convolucionales

Estas capas aplican filtros a regiones locales de la entrada, permitiendo la detección de patrones visuales como bordes, texturas y formas

Funcionamiento de la red

Capas Lineales

Cada neurona en una capa lineal está conectada a todas las neuronas de la capa anterior, aplicando una transformación lineal a los datos de entrada.

$$y = Wx + b$$

Donde:

- y es la salida.
- W es la matriz de pesos.
- x es el vector de entrada.
- b es el sesgo.

Capas convolucionales

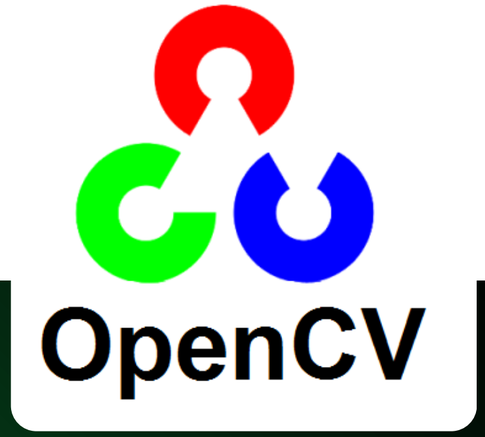
Estas capas aplican filtros a regiones locales de la entrada, permitiendo la detección de patrones visuales como bordes, texturas y formas

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n) K(i-m, j-n)$$

Donde:
S(i,j) es el valor en la posición (i,j) de la salida.
I es la entrada.
K es el kernel o filtro.

Implementación en python

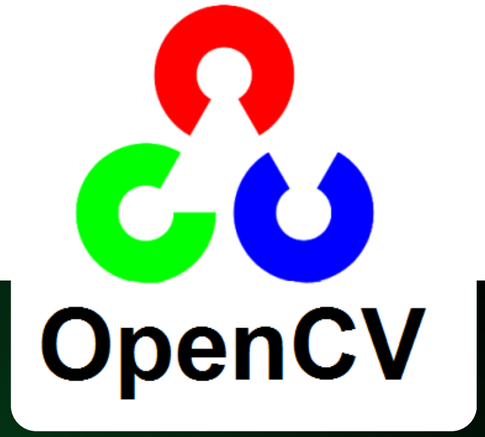
Implementación en python



Opencv

Procesamiento de imágenes

Implementación en python



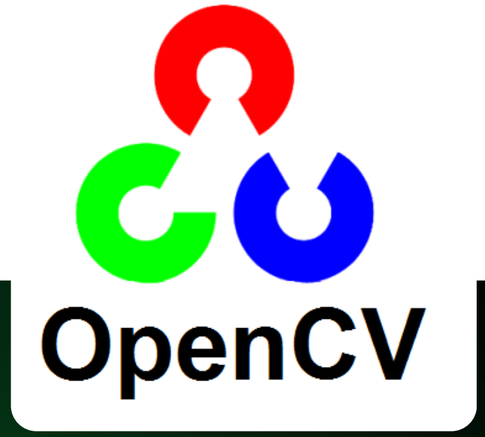
Opencv

Procesamiento de imágenes

Numpy

Optimización de cómputo para el tratamiento de arrays y matrices

Implementación en python



Opencv

Procesamiento de imágenes

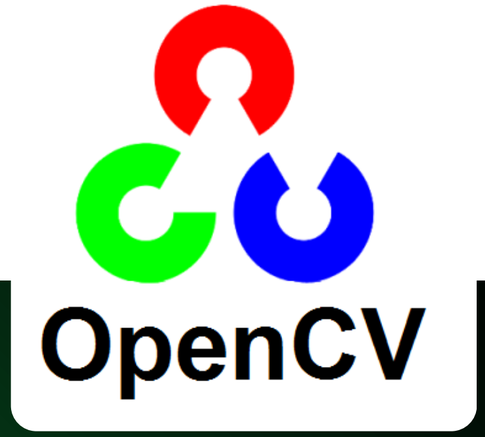
Numpy

Optimización de cómputo para el tratamiento de arrays y matrices

Pytorch

Construcción de redes neuronales totalmente personalizables

Implementación en python



Opencv

Procesamiento de imágenes

Numpy

Optimización de cómputo para el tratamiento de arrays y matrices

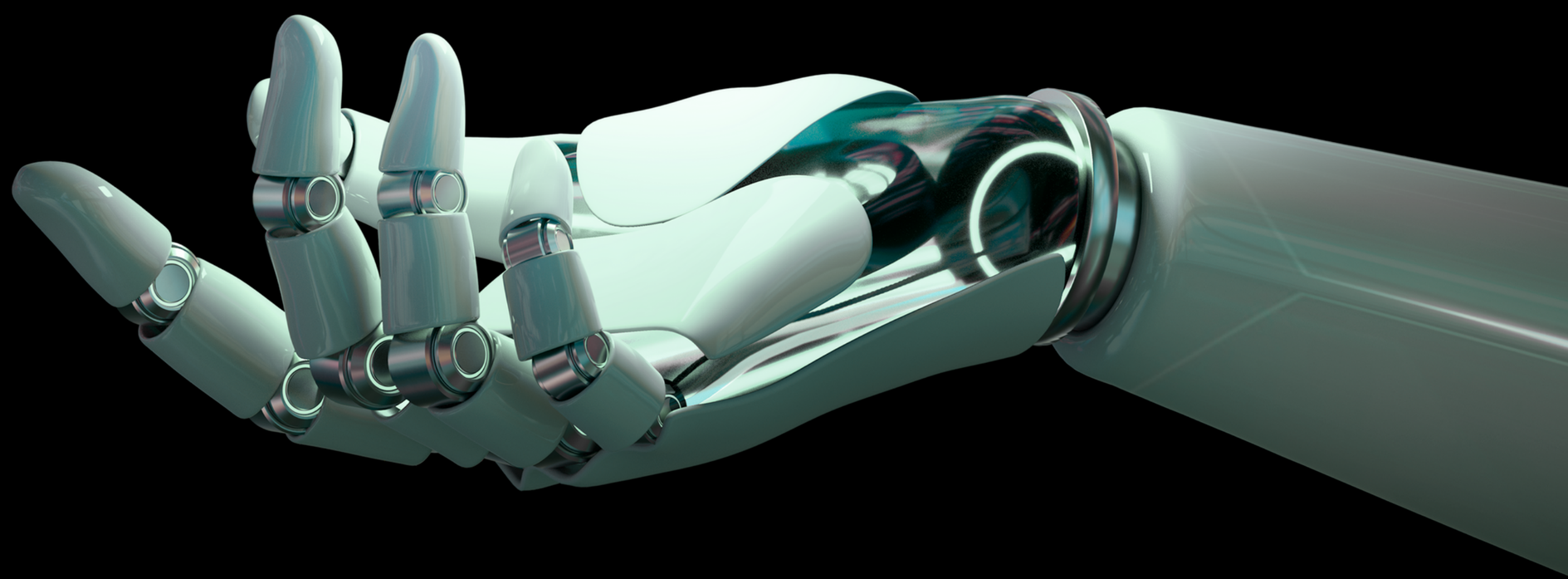
Pytorch

Construcción de redes neuronales totalmente personalizables

Sklearn

Herramientas para construcción y uso de datasets asi como para análisis de resultados.
(En nuestro caso)

IV.Optimizaciones

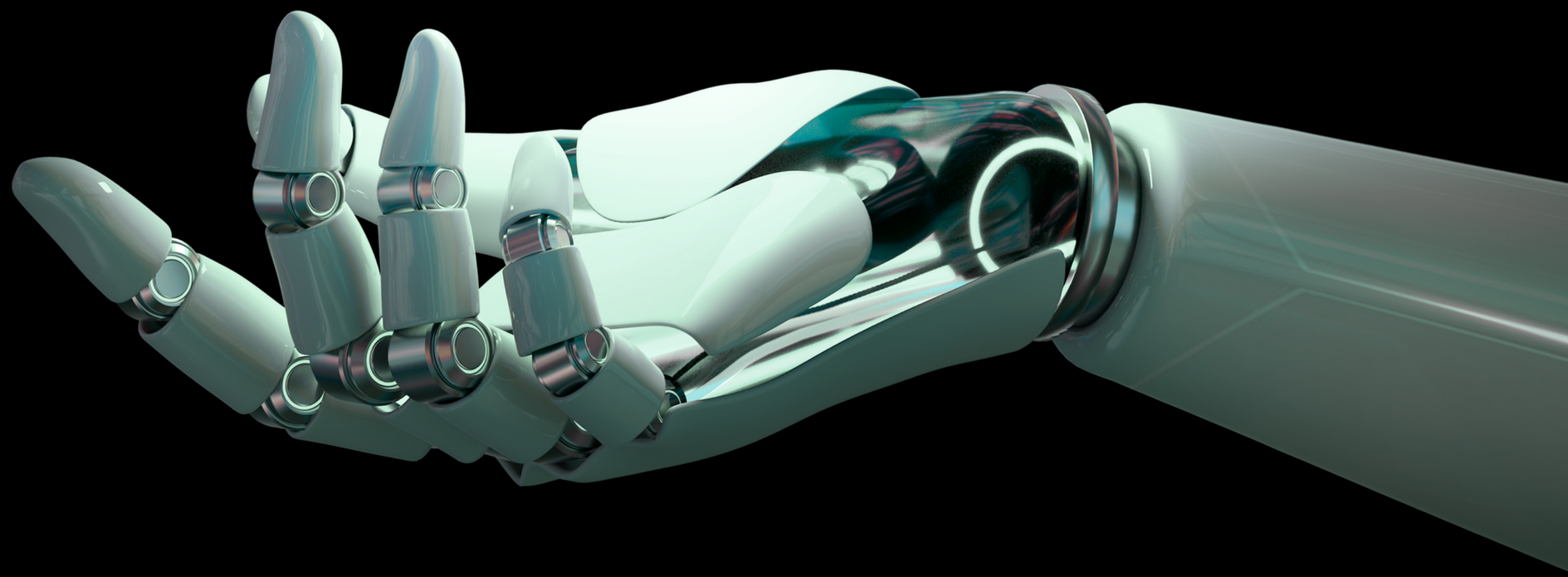


IV.Optimizaciones

Capas convolucionales:

- Kernel de 3x3
- Características del orden de 16*n
- Padding de 1 unidad

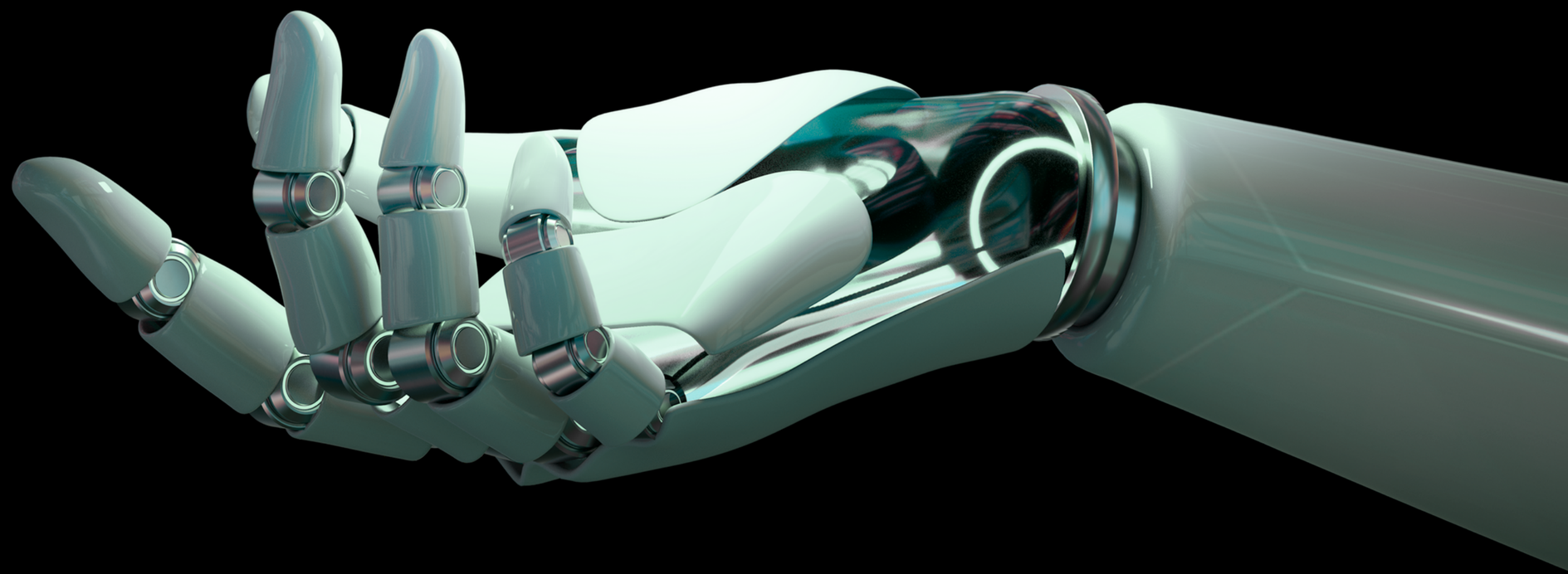
*n = posición de la capa convolucional en la red neuronal



IV.Optimizaciones

Capas de pooling

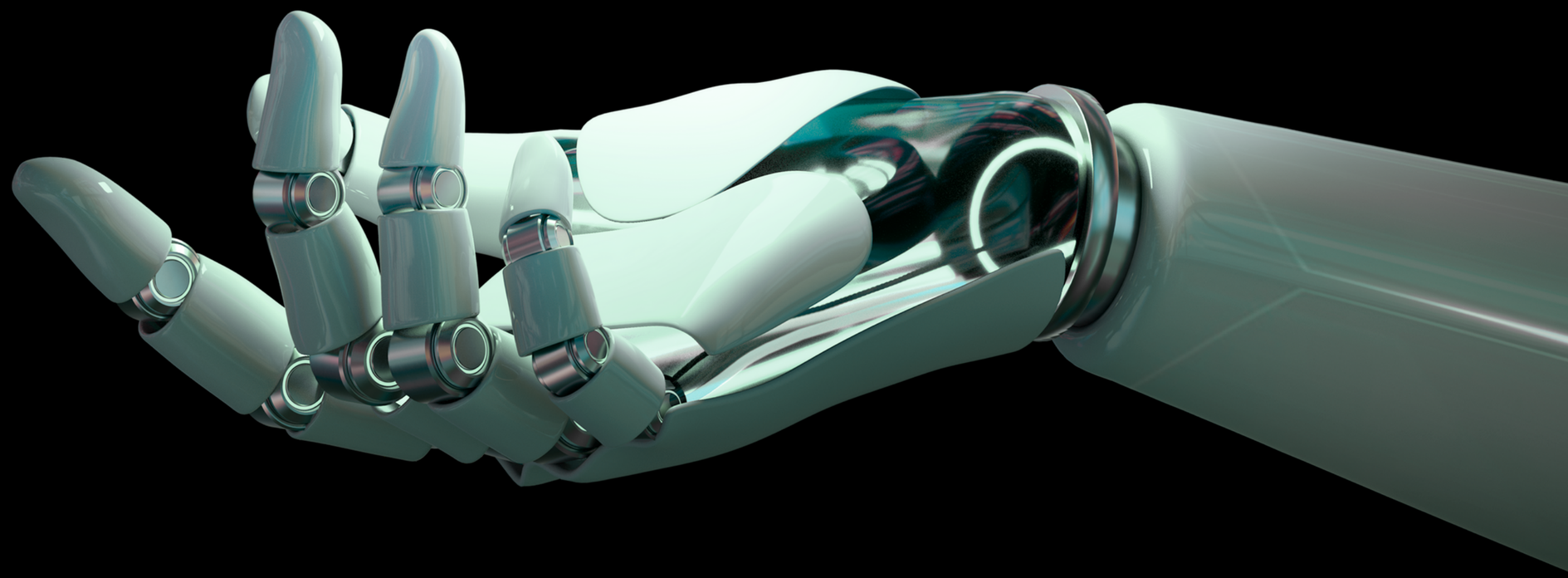
- Kernel de 2x2
- Desplazamiento/stride de 2



IV.Optimizaciones

Capas Lineales

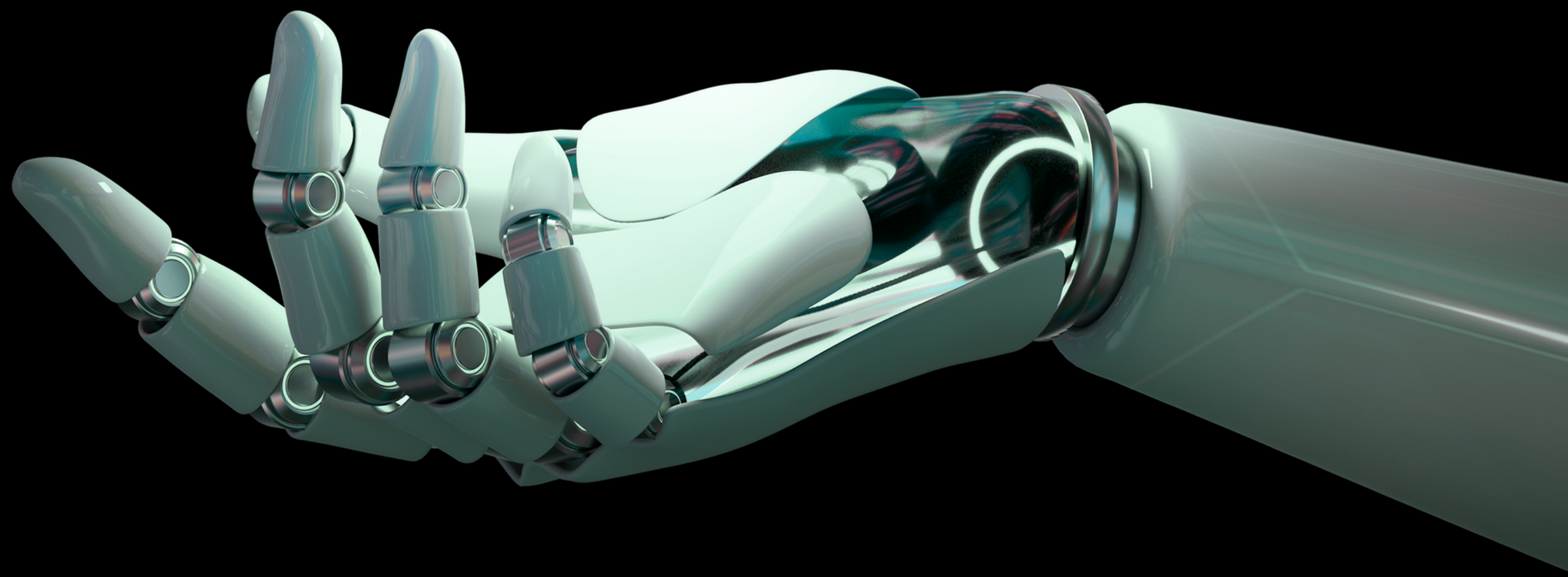
- 2 capa oculta de 512 neuronas
- Función de activación relu
- Se usa después de que los datos pasen por las capas convolucionales y de pooling



IV.Optimizaciones

Función de activación

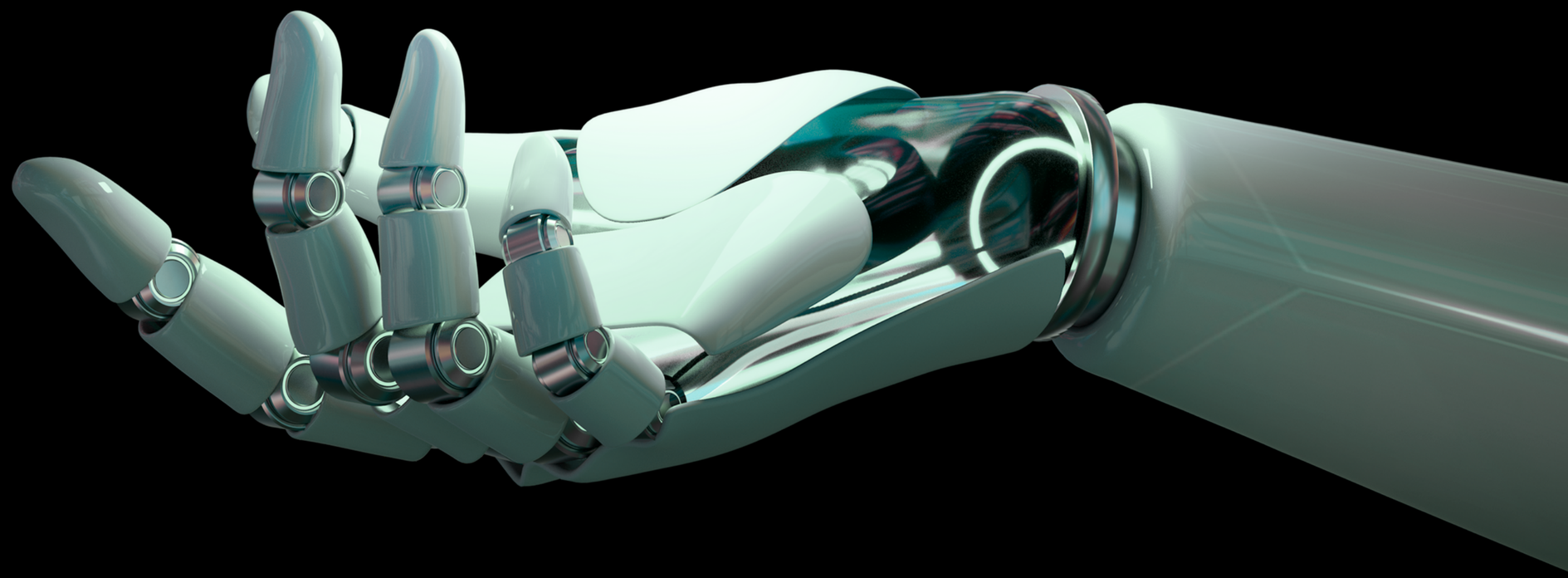
- Relu
- Destaca por su buen funcionamiento para la gran mayoria de redes
- Se usa después de que los datos pasen por las capas convolucionales y por la capa oculta lineal



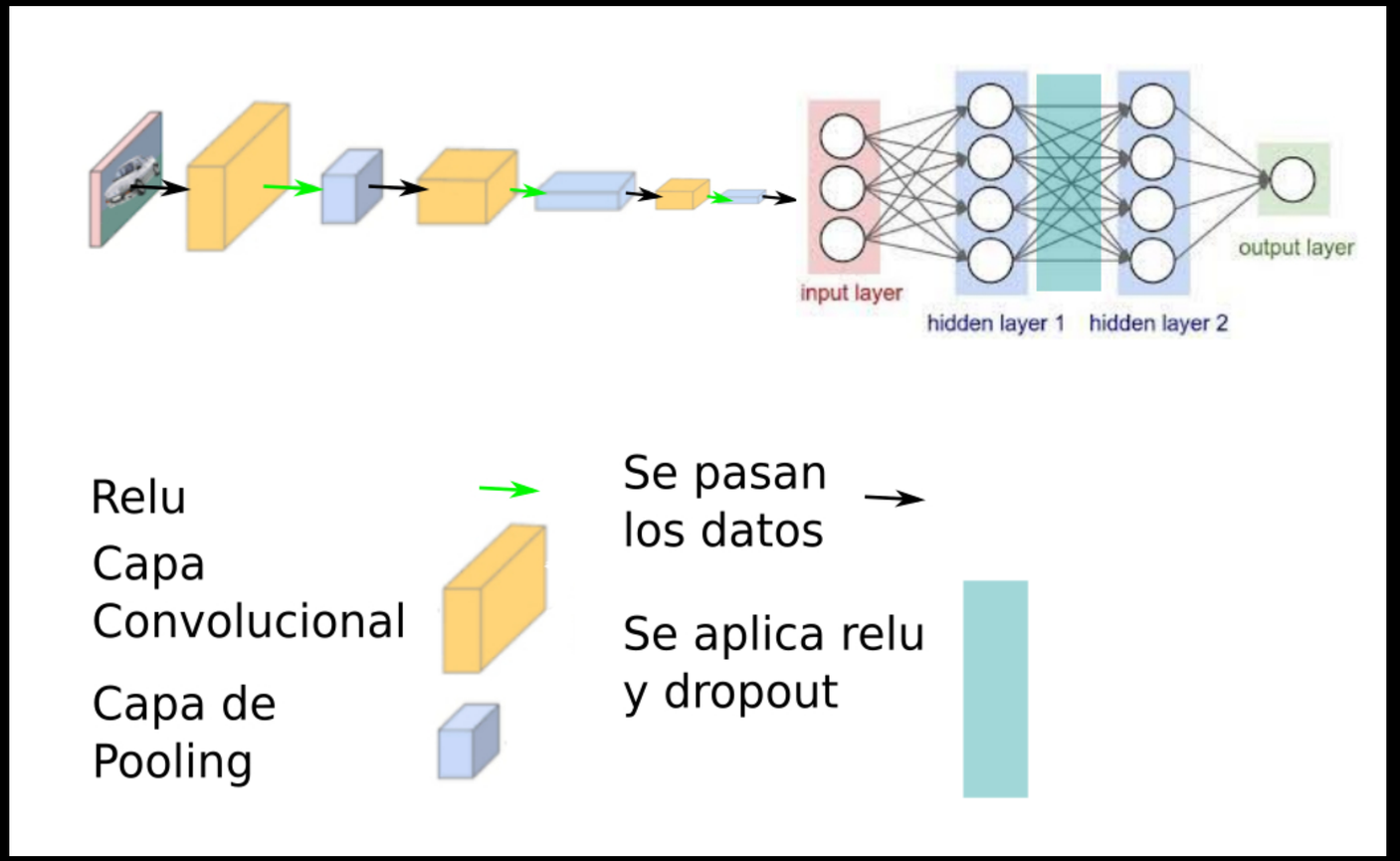
IV.Optimizaciones

Dropout/apagado

- Aumenta la robustez del modelo
- Apaga ciertas neuronas siguiendo una distribución de probabilidad de Bernoulli.
- Lo usamos solo una vez en nuestro caso con una probabilidad de 0.4 despues de pasar los datos por la primera capa lineal oculta



IV.Optimizaciones



Datasets probados

Datasets probados

Datasets probados

Datasets probados

2 Razas de Perros

2 Razas de Perros

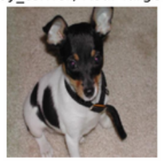
Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: Golden_retriever

Prediction: toy_terrier, Real Target: toy_terrier



Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: toy_terrier, Real Target: toy_terrier

2 Razas de Perros

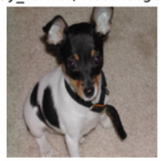
Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: Golden_retriever

Prediction: toy_terrier, Real Target: toy_terrier



Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: Golden_retriever, Real Target: Golden_retriever

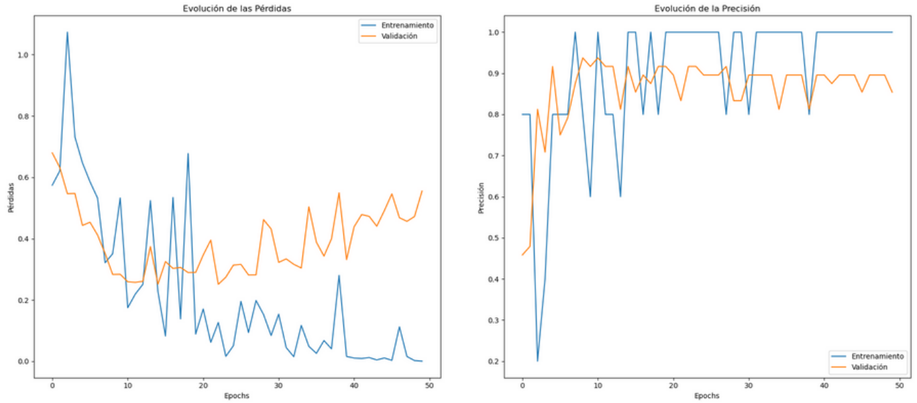
Prediction: toy_terrier, Real Target: toy_terrier

Prediction: Golden_retriever, Real Target: Golden_retriever

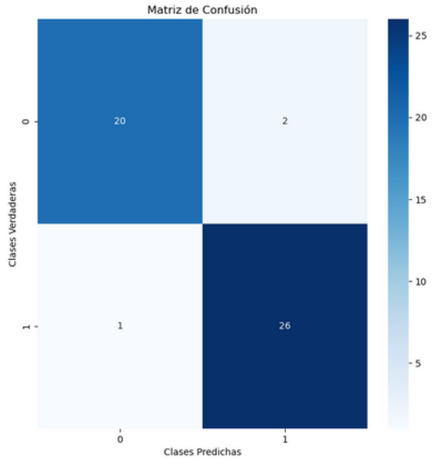
Prediction: toy_terrier, Real Target: toy_terrier

Tasa de acierto

Pérdidas y Precisión



Matriz de confusión



3 Razas de Perros

3 Razas de Perros

Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: Staffordshire_bullterrier, Real Target: Golden_retriever

Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: toy_terrier, Real Target: toy_terrier

Prediction: Staffordshire_bullterrier, Real Target: Golden_retriever

Prediction: Staffordshire_bullterrier, Real Target: Staffordshire_bullterrier

Prediction: Staffordshire_bullterrier, Real Target: Staffordshire_bullterrier

3 Razas de Perros

Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: Staffordshire_bullterrier, Real Target: Golden_retriever

Prediction: Golden_retriever, Real Target: Golden_retriever

Prediction: toy_terrier, Real Target: toy_terrier

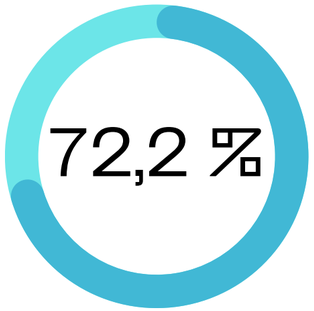
Prediction: toy_terrier, Real Target: toy_terrier

Prediction: Staffordshire_bullterrier, Real Target: Golden_retriever

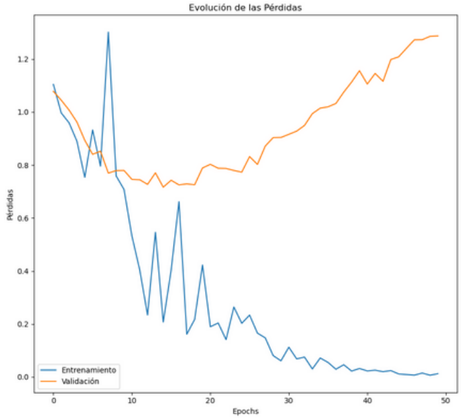
Prediction: Staffordshire_bullterrier, Real Target: Staffordshire_bullterrier

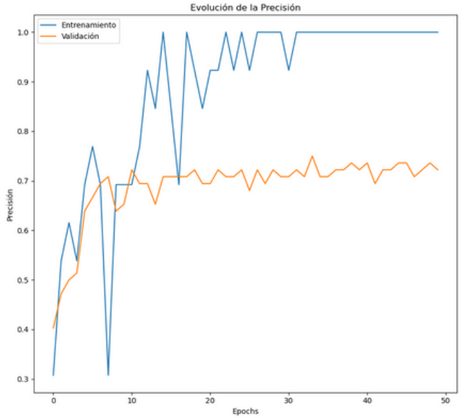
Prediction: Staffordshire_bullterrier, Real Target: Staffordshire_bullterrier

Tasa de acierto

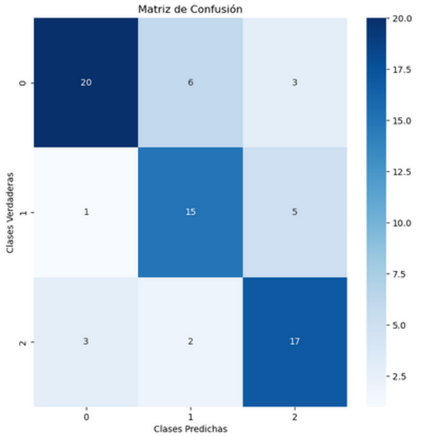


Pérdidas y Precisión





Matriz de confusión



9 Modelos de Coches

9 Modelos de Coches

Pred: octavia, Real: hyundai_i10

Pred: chevrolet_spark, Real: seat_ibiza

Pred: chevrolet_spark, Real: chevrolet_spark

Pred: octavia, Real: seat_ibiza

Pred: toyota_corolla, Real: toyota_corolla

Pred: chevroulet_aveo, Real: chevroulet_aveo

Pred: chevroulet_aveo, Real: chevroulet_aveo

Pred: octavia, Real: octavia

Pred: toyota_corolla, Real: toyota_corolla

Pred: hyundai tucson, Real: polo

9 Modelos de Coches

Pred: octavia, Real: hyundai_i10

Pred: chevrolet_spark, Real: seat_ibiza

Pred: chevrolet_spark, Real: chevrolet_spark

Pred: octavia, Real: seat_ibiza

Pred: toyota_corolla, Real: toyota_corolla

Pred: chevroulet_aveo, Real: chevroulet_aveo

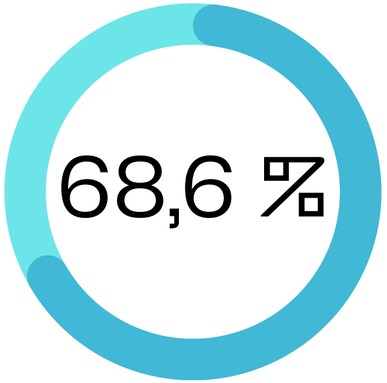
Pred: chevroulet_aveo, Real: chevroulet_aveo

Pred: octavia, Real: octavia

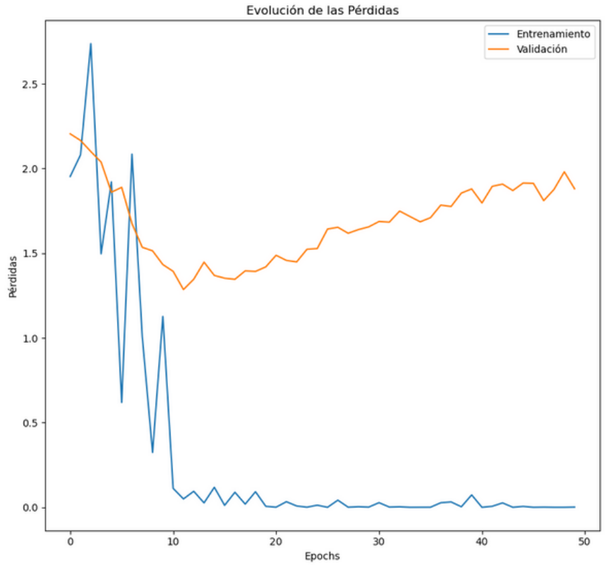
Pred: toyota_corolla, Real: toyota_corolla

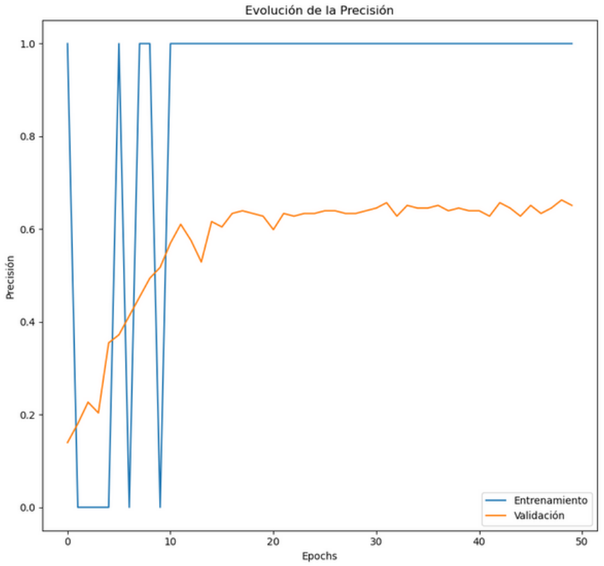
Pred: hyundai tucson, Real: polo

Tasa de acierto



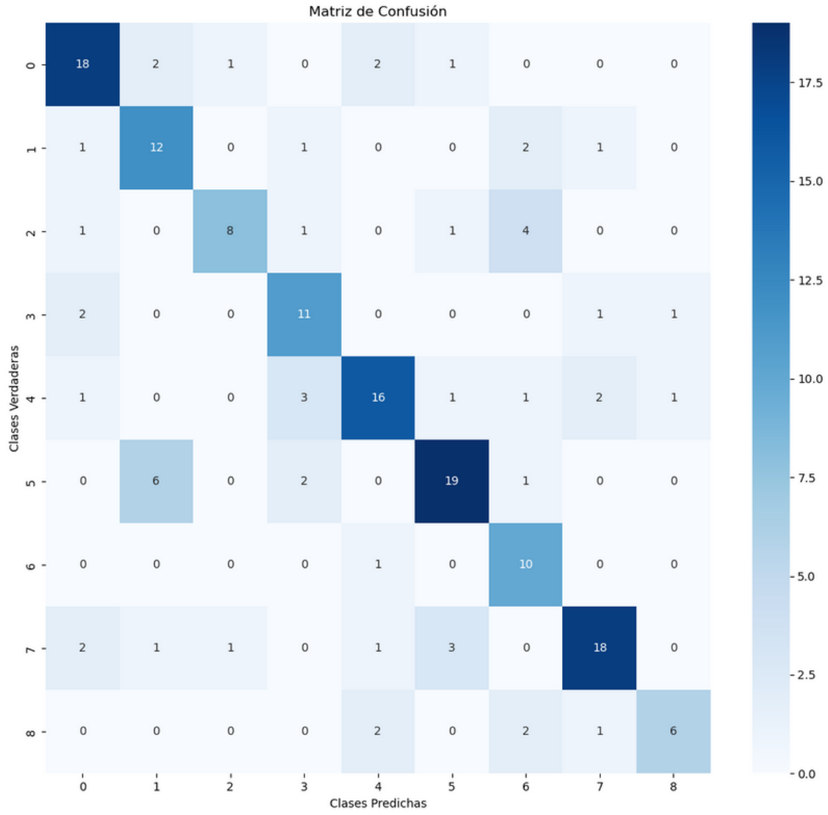
Pérdidas y Precisión





9 Modelos de Coches

Matriz de confusión



102 Tipos de Pokemons

102 Tipos de Pokemons

Pred: cherrim, Real: cherrim

Pred: boldore, Real: boldore

Pred: chandelure, Real: chandelure

Pred: cacnea, Real: cacnea

Pred: blitzle, Real: blitzle

Pred: beldum, Real: beldum

Pred: aromatisse, Real: aromatisse

Pred: archen, Real: archen

Pred: chesnaught, Real: chesnaught

Pred: cacturne, Real: cacturne

102 Tipos de Pokemons

Pred: cherrim, Real: cherrim

Pred: boldore, Real: boldore

Pred: chandelure, Real: chandelure

Pred: cacnea, Real: cacnea

Pred: blitzle, Real: blitzle

Pred: beldum, Real: beldum

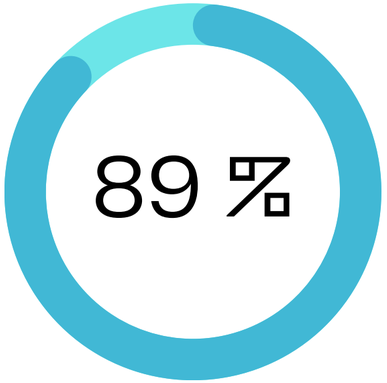
Pred: aromatisse, Real: aromatisse

Pred: archen, Real: archen

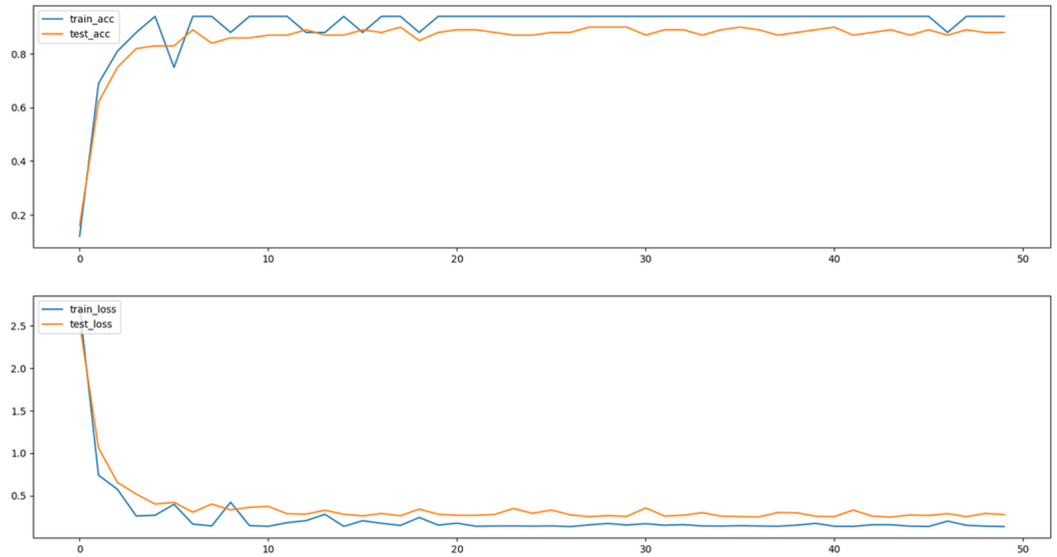
Pred: chesnaught, Real: chesnaught

Pred: cacturne, Real: cacturne

Tasa de acierto



Pérdidas y Precisión



Conclusiones

Conclusiones

Tras los resultados obtenidos de la red neuronal frente a los distintos datasets podemos destacar tres puntos importantes:

Conclusiones

Tras los resultados obtenidos de la red neuronal frente a los distintos datasets podemos destacar tres puntos importantes:

- La complejidad para el modelo aumenta al intentar clasificar elementos muy parecidos, tanto en forma, color, tamaño, etc.

Conclusiones

Tras los resultados obtenidos de la red neuronal frente a los distintos datasets podemos destacar tres puntos importantes:

- La complejidad para el modelo aumenta al intentar clasificar elementos muy parecidos, tanto en forma, color, tamaño, etc.
- La arquitectura es robusta independientemente del dataset (siempre que los datos de entrada sean matrices 2D con n canales, es decir datos de mxnxc dimensiones).

Conclusiones

Tras los resultados obtenidos de la red neuronal frente a los distintos datasets podemos destacar tres puntos importantes:

- La complejidad para el modelo aumenta al intentar clasificar elementos muy parecidos, tanto en forma, color, tamaño, etc.
- La arquitectura es robusta independientemente del dataset (siempre que los datos de entrada sean matrices 2D con n canales, es decir datos de mxnxc dimensiones).
- El desempeño de la red es similar al que tendría un ser humano debido al aumento del error con imagenes que incluyen oclusiones, similitudes entre clases, etc.

Conclusiones

Tras los resultados obtenidos de la red neuronal frente a los distintos datasets podemos destacar tres puntos importantes:

- La complejidad para el modelo aumenta al intentar clasificar elementos muy parecidos, tanto en forma, color, tamaño, etc.
- La arquitectura es robusta independientemente del dataset (siempre que los datos de entrada sean matrices 2D con n canales, es decir datos de mxnxc dimensiones)
- El desempeño de la red es similar al que tendría un ser humano debido al aumento del error con imágenes que incluyen oclusiones, similitudes entre clases, etc

Por estas razones creemos que la arquitectura de red neuronal empleada en este estudio es recomendable para cualquier problema de clasificación de imágenes.

GRACIAS POR VUESTRA ATENCIÓN