## Abstract

Como proyecto final de la asignatura de Machine Learning, se nos planteó el reto de desarrollar un sistema de reconocimiento facial capaz de identificar a nuestros compañeros de clase en una imagen, es decir, reconocer quién aparece en una fotografía.

El abordaje de este desafío se descompone en dos fases: la detección de rostros en imágenes y su posterior reconocimiento. En este proyecto, nos enfocaremos principalmente en la segunda fase, analizando el uso de Redes Neuronales Convolucionales (CNNs) en el reconocimiento facial, con especial atención a las redes siamesas. Estas redes han demostrado un rendimiento efectivo al comparar características faciales y en la verificación de identidad.

A través de la implementación de un modelo siamés, se explorará la capacidad de esta metodología para mejorar la precisión y robustez en el reconocimiento de personas no vistas en el conjunto de entrenamiento. Este enfoque innovador podría abrir nuevas vías para la identificación en entornos educativos, donde la diversidad de rostros y poses puede presentar desafíos únicos.

Los resultados de este proyecto se espera que contribuyan a la comprensión de la efectividad de las redes siamesas en el reconocimiento facial, proporcionando así un análisis crítico que podría ser útil para futuras investigaciones y aplicaciones en el ámbito de la inteligencia artificial y la visión por computadora.


## 1 Introducción

El reconocimiento facial representa un desafío técnico complejo, especialmente en entornos con datos limitados. En este proyecto, nos enfrentamos a la problemática de contar con un conjunto de imágenes reducido, en particular, la ausencia de imágenes de los compañeros de aula durante el entrenamiento. Esto planteó una dificultad central: construir un modelo capaz de identificar a personas que no habían sido incluidas previamente en el entrenamiento, en este informe nos enfocaremos en hacer una analisis sobre el problema cero-shot learning, exigiendo que el sistema generalizara de manera efectiva a nuevos individuos.

Para abordar este desafío, optamos por mantener el enfoque más sencillo posible para la detección de rostros, utilizando el algoritmo Viola-Jones como base para detectar caras en las imágenes. Este algoritmo permitió detectar rostros con precisión, lo que estableció una sólida base para la fase de reconocimiento, donde se pondrían a prueba las capacidades del sistema para identificar a compañeros nuevos sin haber sido entrenado específicamente para ello.

A medida que avanzamos, diseñamos y ajustamos nuevos modelos con el fin de mejorar continuamente los resultados, centrándonos en la simplicidad y en maximizar la capacidad de generalización del sistema. Las mejoras fueron evaluadas en función de los siguientes criterios clave:

- Precisión en la identificación de nuevos individuos: Evaluamos si el sistema era capaz de reconocer a compañeros que no formaban parte del conjunto de entrenamiento inicial. - Eficiencia en la detección y reconocimiento: Medimos la velocidad y la precisión en la detección de rostros mediante el algoritmo Viola-Jones, garantizando que el proceso fuese eficiente y sencillo. - Capacidad de generalización: Analizamos hasta qué punto los nuevos modelos mejoraban la capacidad del sistema para manejar imágenes de personas no vistas previamente, priorizando la generalización sin necesidad de grandes volúmenes de datos. - Comparación iterativa: En cada fase del proyecto, comparamos el desempeño de los nuevos modelos con las versiones anteriores, midiendo las mejoras en precisión, velocidad y generalización.

Con este enfoque iterativo y basado en soluciones sencillas, logramos refinar progresivamente el sistema de reconocimiento facial, superando las limitaciones iniciales impuestas por la falta de datos. Esto nos permitió mejorar la capacidad del modelo para reconocer personas no incluidas en el conjunto de entrenamiento, mientras manteníamos la simplicidad y la eficiencia en el proceso.

## 2 Estado del Arte en Verificación de Identidad Facial

El campo del reconocimiento facial ha avanzado de manera significativa en la última década, impulsado por la evolución de las redes neuronales profundas y la disponibilidad de grandes conjuntos de datos. La verificación de identidad fa-

cial, que busca determinar si un rostro nuevo pertenece a una persona previamente registrada en una base de datos, se beneficia de estos avances. En esta sección, exploraremos las metodologías más recientes y efectivas, describiendo sus características técnicas y arquitectónicas, así como su precisión en diversas condiciones.

### 2.1 FaceNet

Uno de los modelos más influyentes en este campo es **FaceNet**, desarrollado por Google en 2015. FaceNet emplea una arquitectura de red neuronal siamesa que genera embeddings faciales. Esta técnica se basa en la idea de que las imágenes de la misma persona deben estar cercanas entre sí en el espacio de características, mientras que las imágenes de diferentes personas deben estar alejadas. FaceNet utiliza una función de pérdida triplete, que se centra en optimizar la distancia entre un ancla (una imagen de referencia), una imagen positiva (de la misma persona) y una imagen negativa (de otra persona). Este enfoque permite una comparación eficiente utilizando la distancia euclidiana entre embeddings. En el conjunto de datos **LFW** (Labeled Faces in the Wild), FaceNet logró una precisión del 97.3%, destacándose en condiciones de iluminación controladas y variaciones de poses.

### 2.2 ArcFace

Un avance posterior significativo es **ArcFace**, que se basa en una arquitectura de red residual profunda, específicamente **ResNet-100**. La principal innovación de ArcFace es su función de pérdida conocida como **Additive Angular Margin Loss (AM-Softmax)**, que maximiza la separación angular entre los embeddings de diferentes identidades. Este enfoque no solo mejora la discriminación entre clases, sino que también optimiza el rendimiento en condiciones donde los rostros pueden compartir características similares. En pruebas realizadas en **LFW**, ArcFace alcanzó una impresionante precisión superior al 99.8%, así como un 99.63% en el conjunto de datos **MegaFace**, evidenciando su efectividad tanto en escenarios controlados como desafiantes.

### 2.3 CosFace

Otra alternativa destacada es **CosFace**, que introduce el uso de una distancia cosenoidal para mejorar la generalización en la verificación facial. También basado en **ResNet-50**, CosFace implementa una variación de la función de pérdida AM-Softmax, centrada en la distancia entre embeddings faciales. Los resultados obtenidos en el conjunto de datos LFW revelaron una precisión del 99.73%, mientras que en MegaFace logró un 95.1%. Esto destaca la capacidad de CosFace para adaptarse a la variabilidad de los datos y su robustez en condiciones cambiantes.

### 2.4 VGGFace2

Por su parte, **VGGFace2** se centra en la robustez frente a variaciones faciales, como diferencias en pose, iluminación y expresión. Este modelo utiliza una arquitectura que combina **ResNet-50** y **SENet-50** (Squeeze-and-Excitation Networks). SENet introduce un mecanismo de atención que ajusta el peso de cada canal de características, permitiendo que el modelo se concentre en las características más relevantes para la verificación facial. VGGFace2 ha alcanzado una precisión del 99.2% en LFW y ha mostrado resultados competitivos en entornos no controlados, lo que subraya su aplicabilidad en escenarios del mundo real.

### 2.5 CurricularFace

Más recientemente, **CurricularFace** ha introducido un enfoque de aprendizaje jerárquico que optimiza el rendimiento en la clasificación de imágenes difíciles. Utilizando una arquitectura basada en **ResNet-100**, este modelo presenta imágenes más simples en las primeras etapas del entrenamiento y gradualmente introduce imágenes más complejas. Este método permite que la red aprenda a resolver primero los casos fáciles, ajustándose posteriormente a los más desafiantes. CurricularFace ha mostrado precisiones de hasta 99.8% en LFW, siendo especialmente útil en situaciones donde la calidad de las imágenes puede variar significativamente.

### 2.6 Vision Transformers y Reconocimiento Facial enmascarado

En los últimos años, hemos visto la creciente popularidad de los **Transformers** en el ámbito de la visión por computadora. Modelos como **Vision Transformers (ViT)** y **Swin Transformer** han comenzado a ser explorados para tareas de reconocimiento facial. Estas arquitecturas permiten procesar las imágenes como secuencias de parches, mejorando la capacidad de capturar relaciones espaciales de largo alcance, crucial en el análisis facial. Este enfoque se complementa con

el reconocimiento de rostros enmascarados, impulsado por la pandemia de COVID-19, que ha llevado al desarrollo de modelos como **Masked ArcFace** y **CurricularFace** adaptados para reconocer rostros parcialmente cubiertos. **Masked ArcFace** ha demostrado mantener una precisión del 91.5% en el reconocimiento facial enmascarado, evidenciando su eficacia en situaciones desafiantes donde los rostros están parcialmente ocultos.

## 3 Objetivos

El objetivo principal de este proyecto es desarrollar un sistema de reconocimiento facial eficiente, capaz de identificar correctamente a personas no incluidas en el conjunto de entrenamiento, utilizando un enfoque sencillo y con datos limitados. Los objetivos específicos son:

- **Desarrollar un modelo de detección y reconocimiento facial**: Utilizar el algoritmo Viola-Jones para la detección de rostros y anlizar difernetes técnicas para el reconocimiento.
- **Evaluar la capacidad del modelo para generalizar**: Medir la capacidad del sistema para identificar correctamente individuos no vistos durante el entrenamiento.
- **Mejorar la precisión y eficiencia del sistema**: Optimizar el balance entre precisión en la identificación y eficiencia computacional. experimentando con estas

## 4 Metodologías

En este proyecto de reconocimiento facial, implementaremos y evaluaremos diversos enfoques que van desde los más sencillos hasta los más complejos. A lo largo del proceso, experimentaremos con estas técnicas con el objetivo de mejorar la precisión y la eficiencia del sistema.

### 4.1 Detección de rostros con Viola-Jones

En las primeras etapas del proyecto, utilizamos el algoritmo **Viola-Jones** para la detección de rostros. Este algoritmo transforma las imágenes a escala de grises para reducir la cantidad de datos a procesar, y emplea características *Haar* para identificar rostros dentro de una ventana que se desplaza por la imagen.

Para mejorar la precisión en el reconocimiento, aplicamos *data augmentation*,Esta técnica ayudó a aumentar la diversidad del conjunto de entrenamiento sin requerir más imágenes reales, lo que permitió obtener mejores resultados en la predicción posterior.

### 4.2 Reconocimiento con LBPH (Local Binary Patterns Histogram)

El algoritmo **LBPH (Local Binary Patterns Histogram)** fue utilizado para el reconocimiento facial. Este enfoque genera una representación intermedia de la imagen en la que se resaltan las características del rostro. Posteriormente, se divide la imagen en una cuadrícula y se calcula un histograma para cada celda, el cual describe la distribución de los valores de intensidad de los píxeles. Finalmente, los histogramas se concatenan para formar un descriptor único que se compara con los histogramas previamente entrenados.

El proceso de reconocimiento consistía en los siguientes pasos:

1. Captura de la imagen o video a través de una cámara.
2. Aplicación de Viola-Jones para detectar el rostro en la imagen.
3. Uso de LBPH para extraer el histograma de la cara detectada.
4. Comparación del histograma con los modelos previamente entrenados.
5. Retorno del ID correspondiente al rostro más parecido.

Aunque el sistema era relativamente rápido, la precisión no superaba el 60%, incluso con técnicas de *data augmentation*. Esto nos llevó a explorar otros enfoques más avanzados.

### 4.3 Redes Convolucionales Siamesas

En un intento por mejorar la precisión del modelo, probamos redes convolucionales siamesas. A diferencia de los modelos tradicionales, las redes siamesas no clasifican imágenes en etiquetas específicas, sino que miden la distancia entre los *embeddings* de dos imágenes. El flujo del proceso es el siguiente:

1. **Procesamiento de Imágenes**: Dos imágenes son procesadas simultáneamente por redes con la misma arquitectura.

2. **Obtención de Embeddings**: Se generan los *embeddings* de ambas imágenes a través de la red.

3. **Cálculo de Distancia**: Se calcula la distancia euclidiana entre los *embeddings*. Cuanto menor sea la distancia, más similares son las imágenes.

Con el objetivo de mejorar nuestro sistema, experimentamos con diferentes arquitecturas para las redes, logrando una pequeña mejora.Sin embargo, a pesar de que este enfoque parecía prometedor, en la práctica presentó varios inconvenientes. Su demanda computacional resultaba alta, lo que lo hacía poco práctico para equipos con recursos limitados. Esto complicó la experimentación con la arquitectura de la red.

Por lo tanto, decidimos intentar utilizar un modelo preentrenado como base para la red siamesa, aplicando *transfer learning* para ajustarlo específicamente a nuestra tarea. Para ello, utilizamos la arquitectura *ResNet-100*. No obstante, la demanda computacional de este modelo seguía siendo elevada, lo que nos llevó a explorar otros enfoques.

### 4.4 Transición hacia FaceNet

Finalmente, optamos por implementar **FaceNet**, un modelo de reconocimiento facial basado en *embeddings* euclidianos, que requiere solo unas pocas imágenes para reconocer a una persona con alta precisión. FaceNet fue entrenado previamente con más de 140 mil imágenes, permitiendo una precisión del 99.2% con solo 2 o 3 fotos por persona.

### 4.5 Transición hacia FaceNet

Finalmente, decidimos implementar **FaceNet**, un avanzado modelo de reconocimiento facial basado en *embeddings* euclidianos. A diferencia de otros enfoques, FaceNet es capaz de reconocer a una persona con alta precisión utilizando solo unas pocas imágenes. Gracias a su preentrenamiento con más de 140 mil imágenes, FaceNet alcanza una precisión del 99.2% con apenas 2 o 3 fotos por individuo, lo que lo convierte en una opción robusta y eficiente para tareas de identificación facial.

### 4.6 Transición hacia FaceNet

En la búsqueda de mejorar la eficiencia y precisión de nuestro sistema, exploramos alternativas que se alinearan con el enfoque de aprendizaje con (*zero-shot learning*). Este enfoque es crucial para escenarios donde el modelo debe generalizar a nuevas clases sin haber visto ejemplos durante el entrenamiento. Con este objetivo, optamos por implementar **FaceNet**, un modelo diseñado para generar *embeddings* faciales a partir de distancias euclidianas en un espacio de características, en lugar de depender de una clasificación modelo diseñado para generar *embeddings* faciales a partir de distancias euclidianas en un espacio de características, en lugar de depender de una clasificación directa.

Una de las principales ventajas de FaceNet es su capacidad para realizar tareas de identificación y verificación de rostros sin requerir que el modelo sea entrenado con las mismas personas que se desea reconocer. En vez de entrenar el modelo con estas nuevas imágenes, utilizamos las imágenes de referencia únicamente para calcular los *embeddings* faciales y evaluar la similitud entre ellos, lo que lo hace altamente adecuado para el problema de reconocimiento de identidad con datos previamente no vistos.

FaceNet fue preentrenado con un extenso conjunto de datos de más de 140 mil imágenes, lo que le confiere una capacidad robusta para generar representaciones discriminativas de rostros, alcanzando una precisión del 99.2%. Este modelo puede funcionar eficientemente con tan solo 2 o 3 imágenes de referencia por individuo, facilitando así la identificación de personas en situaciones donde no se cuenta con grandes volúmenes de datos específicos para cada clase.

## 5 Experimentación

En la implementación de un sistema de reconocimiento facial utilizando una red neuronal siamesa, se realizaron diversas experimentaciones con el objetivo de optimizar la precisión y la capacidad de generalización del modelo. Estas experimentaciones se centraron en la correcta creación y estructuración del conjunto de datos, la metodología de entrenamiento y el ajuste de hiperparámetros, para lograr un modelo robusto y consistente.

Las redes siamesas requieren pares de imágenes que representen relaciones entre una imagen de referencia, denominada **ancla** (*anchor*), y otra imagen, que puede ser de la misma persona (**par positivo**) o de una persona diferente (**par negativo**).

Estos pares se etiquetan como 1 (par positivo) o 0 (par negativo), formando las siguientes tuplas:

- $\langle anchor, positive, 1 \rangle$: Ambas imágenes corresponden a la misma persona.
- $\langle anchor, negative, 0 \rangle$: Las imágenes pertenecen a personas distintas.

El modo en que se estructuran y seleccionan estos pares tiene un impacto crítico en el rendimiento del modelo, como se evidenció en las siguientes iteraciones experimentales.

### 5.1 Primera Iteración: Entrenamiento con Datos Limitados

En esta fase inicial, se diseñó un conjunto de datos limitado, buscando que el modelo aprendiera a identificar una única persona. Para ello, se crearon pares de imágenes donde las anclas y las imágenes positivas provenían de una sola identidad, mientras que las imágenes negativas se seleccionaron del conjunto LFW (*Labeled Faces in the Wild*). Se generaron aproximadamente 3000 pares, distribuidos equitativamente entre positivos y negativos.

Este enfoque resultó ser ineficaz. El modelo no logró generalizar adecuadamente, presentando una tasa de precisión extremadamente baja. Al entrenarse con un conjunto de datos limitado y centrado en una única identidad, el modelo no aprendió representaciones suficientemente discriminativas para reconocer a otras personas. Esta limitación evidenció la falta de robustez del modelo frente a nuevos individuos.

Esta primera iteración destacó la necesidad de aumentar la diversidad del conjunto de datos para mejorar la capacidad de generalización.

### 5.2 Segunda Iteración: Expansión y Selección Aleatoria del Conjunto de Datos

Aprendiendo de la limitación anterior, se amplió el tamaño del conjunto de datos a 14,000 pares, seleccionados de un conjunto de aproximadamente 100,000 imágenes de LFW. La selección incluyó 7000 pares positivos y 7000 pares negativos. Sin embargo, cometimos un error crítico: los pares se seleccionaron de manera completamente aleatoria, sin garantizar que cada imagen de ancla tuviera un par positivo y un par negativo.

Este desbalance en la estructura del conjunto de datos llevó a un entrenamiento inconsistente, lo que resultó en un rendimiento subóptimo. Aunque la tasa de precisión mejoró ligeramente, no alcanzó los niveles esperados. Esta fase resaltó la importancia de mantener un equilibrio adecuado en la selección de pares para cada ancla.

### 5.3 Tercera Iteración: Balance y Reducción del Dataset

En la tercera iteración, se corrigieron los errores de las fases anteriores. Se redujo el conjunto de datos a 6000 pares, asegurando que cada imagen de ancla estuviera asociada tanto a un par positivo como a un par negativo. Este equilibrio permitió que el modelo aprendiera a discriminar de manera más efectiva entre imágenes de la misma persona y de personas diferentes.

La mejora en la estructura del conjunto de datos condujo a una significativa mejora en la capacidad de generalización del modelo. Los resultados de validación mostraron un aumento considerable en la precisión, confirmando que un conjunto de datos equilibrado y bien estructurado es esencial para entrenar redes siamesas de manera efectiva.

### 5.4 Lecciones Aprendidas sobre la Selección del Dataset

A lo largo de estas iteraciones, se extrajeron varias lecciones importantes sobre la selección del conjunto de datos:

1. **Balance en la Estructura del Dataset**: El equilibrio entre pares positivos y negativos es crucial. Cada imagen de ancla debe tener tanto un par positivo como un par negativo para garantizar que el modelo aprenda a discriminar correctamente entre clases.

2. **Diversidad de los Datos**: La falta de diversidad en los datos limita severamente la capacidad de generalización del modelo. Un conjunto de datos más diverso mejora la representación de diferentes identidades y reduce el riesgo de sobreajuste.

3. **Selección Estratégica de Pares**: La selección aleatoria puede llevar a conjuntos de datos desbalanceados y no representativos. Es fundamental adoptar un enfoque más estratégico en la creación de pares para asegurar un entrenamiento eficaz.

4. **Tamaño del Dataset vs. Calidad de los Pares**: Un mayor tamaño de conjunto de datos no garantiza mejores resultados si la

calidad y el balance de los pares no son adecuados. Un conjunto más pequeño pero bien estructurado puede superar a uno grande y desbalanceado.

### 5.5 Experimentación con Hiperparámetros: Tasa de Aprendizaje y Funciones de Pérdida

Además de la selección de datos, realizamos una exhaustiva experimentación con hiperparámetros, centrándonos en la tasa de aprendizaje y las funciones de pérdida.

#### 5.5.1 Tasa de Aprendizaje

Inicialmente, probamos una tasa de aprendizaje de $10^{-4}$, que es comúnmente recomendada en redes profundas. Sin embargo, observamos que el modelo tardaba en converger y no alcanzaba niveles de precisión aceptables. Posteriormente, evaluamos tasas de $10^{-3}$ y $10^{-5}$, pero tampoco dieron mejoras significativas. Finalmente, se encontró que la tasa de aprendizaje de $10^{-2}$ era la más adecuada, permitiendo una convergencia rápida y estable, con una precisión superior. Este valor se adoptó debido a su equilibrio entre velocidad de entrenamiento y rendimiento.

#### 5.5.2 Comparativa de Funciones de Pérdida

Las funciones de pérdida juegan un rol crucial en la optimización del aprendizaje. Se evaluaron tres funciones de pérdida en el contexto de verificación de identidad facial:

- **Triplet Loss**: Optimiza la distancia entre la imagen de ancla y una positiva, mientras maximiza la distancia entre el ancla y una negativa. Genera representaciones altamente discriminativas, pero la selección del margen es compleja.
- **Contrastive Loss**: Minimiza la distancia euclidiana entre pares positivos y la maximiza entre pares negativos. Es más simple que *Triplet Loss*, pero menos eficaz en conjuntos de datos donde las clases son difíciles de separar.
- **Binary Cross-Entropy (BCE) Loss**: Evalúa la probabilidad de que un par pertenezca a la misma clase. Aunque es simple, no optimiza explícitamente las distancias entre los *embeddings*.

| Función de Pérdida | Test Same Person | Test Different Person |
|---|---|---|
| **Triplet Loss** | 94.5% | 70-75% |
| **Contrastive Loss** | 91.8% | 70-75% |
| **Binary Cross-Entropy** | 89.7% | 60-65% |

Table 1: Comparativa de funciones de pérdida con sus precisiones en diferentes conjuntos de prueba.

#### 5.5.3 Conclusión sobre la Mejor Función de Pérdida

La **Triplet Loss** se destacó como la más eficaz para la tarea de verificación facial debido a su capacidad para generar representaciones altamente discriminativas, especialmente cuando se combina con técnicas como *hard negative mining*.

## 6 Conclusiones y Futuras Mejoras

Este proyecto mostró la importancia de una correcta selección y balance de los datos en el entrenamiento de redes siamesas, así como la optimización de hiperparámetros clave como la tasa de aprendizaje y la función de pérdida. En el futuro, se explorarán modelos más avanzados, como redes preentrenadas con técnicas de *transfer learning*, y se buscará mejorar la diversidad y calidad de los datos utilizados en el entrenamiento.

## 7 Estadísticas

A continuación se presentan diversas imágenes que ilustran las estadísticas obtenidas durante el proceso de entrenamiento y evaluación del modelo de reconocimiento facial. Estas visualizaciones ayudan a comprender el rendimiento y la efectividad del modelo.

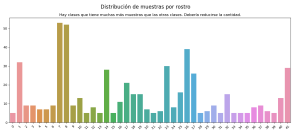

Figure 1: Distribución del dataset utilizado para el entrenamiento.

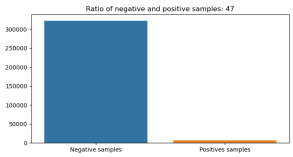


Figure 2: Ejemplo de una muestra del dataset.

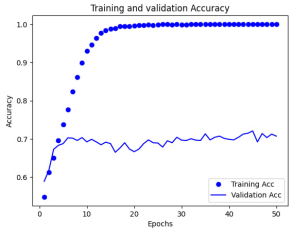


Figure 3: Resultados utilizando 4 capas convolucionales con contrastive loss.

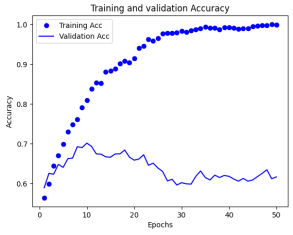


Figure 4: Gráfica de precisión obtenida con el optimizador Adam.

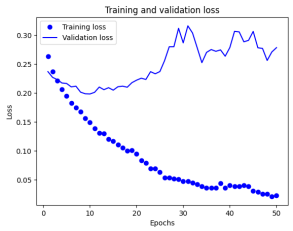


Figure 5: Resultados obtenidos con el optimizador Adam.

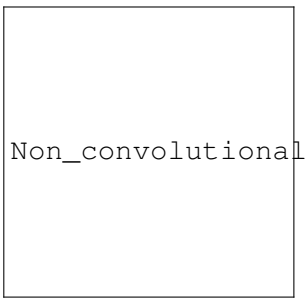


Figure 6: Resultados de una arquitectura no convolucional en términos de precisión.

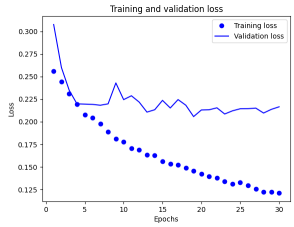


Figure 7: Gráfica de resultados de una arquitectura no convolucional.

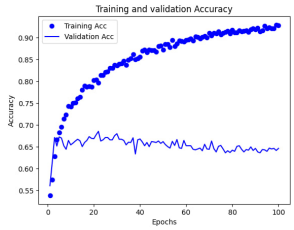


Figure 8: Resultados de una arquitectura no convolucional con un mayor número de épocas.

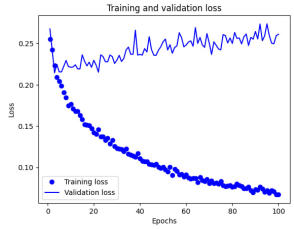


Figure 9: Resultados adicionales de una arquitectura no convolucional con un mayor número de épocas.