MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Universidad del Rosario | Escuela de Ingeniería, Ciencia y Tecnología

Proyecto Final: DigitEye
Clasificación de Imágenes

Diryon Yonith Mora Romero
Laura Valentina Gonzalez Rodriguez

Prof. Yiby Karolina Morales Pinto

Aprendizaje Automático de Máquina
Matemáticas Aplicadas y Ciencias de la Computación
Universidad del Rosario
Mayo 2023

1. Antecedentes

La clasificación de imágenes es una labor importante en el campo del aprendizaje automático y la inteligencia artificial. Teniendo diversas aplicaciones, desde la detección de objetos en la seguridad y la vigilancia, hasta la identificación de enfermedades en la medicina. Crucial en el desarrollo de tecnologías emergentes como los vehículos autónomos y los drones. Sin embargo, con el surgimiento de las redes neuronales y el deep learning, este problema ha vuelto a tomar relevancia, con mejorar cruciales gracias a estas técnicas.

El reconocimiento de imágenes es una tarea difícil debido a la gran variabilidad que existe en las imágenes del mundo real. Esto se debe a la presencia de diferentes fuentes de variabilidad, como cambios de iluminación, orientación, escala, deformación, entre otros factores. Por esta razón, el desarrollo de un clasificador de imágenes preciso y robusto sigue siendo un reto para la comunidad científica.

En 2006, la falta de conjuntos de datos etiquetados fue un obstáculo importante para mejorar los algoritmos de aprendizaje profundo en la visión por computadora. Para abordar este problema, Fei-Fei Li lideró la creación de ImageNet, el conjunto de datos más grande y popular utilizado para la clasificación de imágenes, que contiene más de 14 millones de imágenes etiquetadas en más de 20,000 categorías. Este esfuerzo fue posible gracias a la participación activa de la comunidad de investigadores. [1]

Particularmente, la arquitectura de redes neuronales convolucionales (CNN) ha sido un gran avance en la visión por computadora, debido a su capacidad para reconocer patrones en las imagenes. La red neuronal convolucional ResNet-18 se caracteriza por su profundidad y su capacidad para superar el problema del decaimiento del gradiente. A diferencia de las arquitecturas anteriores, como AlexNet, que utilizaban menos capas, ResNet-18 consta de 18 capas. Estas capas se componen de bloques residuales, que contienen conexiones que saltan capas y permiten el flujo directo de la información. Esto ayuda a mitigar el problema del

decaimiento del gradiente y permite entrenar redes más profundas de manera más efectiva. [2]

Además de la clasificación de imágenes, las redes neuronales convolucionales también han tenido un gran impacto en la detección temprana de enfermedades y trastornos. Un estudio de 2018 demostró que un algoritmo de CNN puede detectar con precisión el cáncer de piel maligno con una precisión del 91 %, lo que es comparable a la precisión de los dermatólogos en la detección de cáncer de piel. [3] En otros casos, se han utilizado algoritmos de CNN en la detección de enfermedades neurológicas, como el Parkinson, a través del análisis de imágenes de resonancia magnética cerebral. [4] Estos avances muestran el enorme potencial de las redes neuronales convolucionales en la medicina y la salud, y su capacidad para ayudar en la detección temprana y el tratamiento de enfermedades.

En resumen, la clasificación de imágenes es una tarea importante y desafiante en el campo del aprendizaje automático y la inteligencia artificial, y tiene muchas aplicaciones prácticas. Las redes neuronales convolucionales se han convertido en una herramienta esencial para reconocer patrones en imágenes y mejorar la precisión y robustez de los algoritmos. La creación de ImageNet fue un hito importante para el campo, ya que proporcionó un gran conjunto de datos etiquetados para entrenar modelos de aprendizaje profundo.

Además, la capacidad de las redes neuronales convolucionales para detectar enfermedades ha demostrado su potencial para mejorar la atención médica y la salud. Con una precisión comparable a la de los profesionales médicos, los algoritmos de CNN se han utilizado en la detección temprana de enfermedades como el cáncer de piel y el Parkinson, lo que demuestra su capacidad para complementar y mejorar la precisión de los diagnósticos médicos. Se espera que los avances continuos en la investigación, sigan mejorando la precisión y eficiencia en la clasificación de imágenes, lo que tendrá un impacto importante en la vida cotidiana.

2. Definición del problema

El problema a resolver es la clasificación de imágenes en diferentes clases. El objetivo de este proyecto es desarrollar un programa que pueda identificar con precisión la categoría en una imagen dada. La motivación personal para llevar a cabo este proyecto radica en el interés por explorar técnicas de aprendizaje automático aplicadas a la visión por computadora, y en la posibilidad de contribuir a la solución de problemas prácticos mediante el desarrollo de un clasificador de imágenes preciso y robusto.

Por lo tanto, el desafío es desarrollar un modelo de aprendizaje automático que pueda generalizar a partir de los datos de entrenamiento, clasificando con precisión las imágenes nuevas. Debe ser capaz de identificar características relevantes y descartar las irrelevantes, y debe ser capaz de adaptarse a las variaciones en las imágenes dentro de cada clase.

3. Descripción de la solución

Para lograr este objetivo, se utilizará un enfoque de aprendizaje automático utilizando redes neuronales convolucionales, que han demostrado ser altamente efectivas en la clasificación de imágenes en diversos estudios. Después de obtener la mejor arquitectura para la CNN, se comparará su rendimiento con el modelo de ResNet-18.

La estrategia propuesta es apropiada, es cuantificable y medible mediante la precisión de la clasificación. El uso de una CNN es una técnica de vanguardia en el campo de la visión por computadora y ha demostrado un alto nivel de precisión en tareas de clasificación de imágenes similares. La implementación de la CNN se realizará utilizando la biblioteca de aprendizaje profundo PyTorch, que proporciona una plataforma de desarrollo eficiente y escalable para la implementación de redes neuronales convolucionales.

4. Datos

El conjunto de datos utilizado en el proyecto, es el conjunto de datos CIFAR-10, el cual consta de 60000 imágenes en color de 32x32 píxeles distribuidas en 10 clases, con 6000 imágenes por clase. Estas clases son: aviones (0), automóviles (1), pájaros (2), gatos (3), ciervos (4), perros (5), ranas (6), caballos (7), barcos (8) y camiones (9). De estas, 50000 imágenes se utilizarán para entrenamiento, 5000 para pruebas y 5000 para validación. El conjunto de pruebas contiene exactamente 1000 imágenes seleccionadas aleatoriamente de cada clase, mientras que los otros contienen las imágenes restantes en orden aleatorio, aunque algunos conjuntos de entrenamiento pueden contener más imágenes de una clase que de otra.

El conjunto de datos CIFAR-10 fue obtenido originalmente por Alex Krizhevsky, Vinod Nair y Geoffrey Hinton, del departamento de Ciencias de la Computación de la Universidad de Toronto. Este conjunto de datos ha sido utilizado en numerosos estudios de investigación en visión por computadora y aprendizaje profundo, y se considera uno de los conjuntos de datos más populares en esta área. Además, la disponibilidad de un gran número de imágenes etiquetadas y la diversidad de las clases presentes en el dataset permiten entrenar una CNN con alta precisión y obtener resultados cuantificables y medibles. [5]

El objetivo del proyecto es utilizar una red neuronal convolucional (CNN) para clasificar las imágenes del conjunto de datos CIFAR-10 en sus respectivas clases, para detectar las caracteristicas especificas de las imágenes. Se emplearán diversas técnicas para mejorar el rendimiento de la CNN, tales como la utilización de capas de convolución y de pooling, la optimización de hiperparámetros. Comparandolo con el modelo ResNet-18, para mirar su efectividad para predecir nuevos datos.

5. Modelo Empleado

Como se menciono, se realizo un modelo simple de red neuronal convolucional y se comparo con el modelo ResNet-18. Dicho código se encuentra en el repositório GitHub. [6]

5.1. Modelo Simple

La red toma una imagen de entrada de tamaño 32x32 píxeles en formato RGB. La imagen de entrada se pasa a través de varias capas convolucionales definidas por bloques SimpleBlock. Cada bloque SimpleBlock consiste en dos capas convolucionales con una función de activación ReLU seguida de una capa de agrupación máxima (max pooling) y una capa de normalización por lotes (batch normalization). Los bloques SimpleBlock extraen características de las imágenes a medida que se procesan en la red. Después de las capas convolucionales, los datos se aplanan en un tensor unidimensional.

A continuación, se aplican varias capas lineales para realizar la clasificación de las características extraídas. Cada capa lineal está seguida de una función de activación ReLU para introducir no linealidad en los datos. La última capa lineal produce las salidas finales, donde el número de neuronas es igual al número de clases de salida. Se aplica una función de activación LogSoftmax para obtener probabilidades normalizadas de las clases. La salida final de la red es un vector de probabilidades que representa la probabilidad de que la imagen de entrada pertenezca a cada una de las clases de salida.

5.2. Modelo ResNet-18

La red toma una imagen de entrada de tamaño 32x32 píxeles en formato RGB. La imagen de entrada pasa a través de una capa convolucional inicial con un kernel de 3x3 para extraer características básicas. Después de la capa convolucional inicial, se aplica una capa

de normalización por lotes y una función de activación ReLU para introducir no linealidad.

A continuación, se pasan las características extraídas a través de una secuencia de bloques básicos de ResNet. Cada bloque básico consiste en dos capas convolucionales seguidas de una capa de normalización y una función de activación ReLU. Las conexiones residuales permiten que el flujo de gradiente se propague más fácilmente a través de las capas, evitando el desvanecimiento del gradiente. Algunos de los bloques básicos tienen conexiones de atajo (shortcut) para ajustar el tamaño de la entrada o el número de canales, si es necesario. Después de los bloques básicos, se aplica una capa de promedio global (AvgPool2d) para reducir el tensor de características a un tamaño 1x1.

El tensor reducido se pasa a través de una capa lineal (linear) para clasificar las características en las clases de salida. La función de activación utilizada en la capa lineal depende del tipo de problema que se esté resolviendo (por ejemplo, una función softmax para clasificación multiclase). La salida final es un vector de probabilidades que representa la probabilidad de que la imagen de entrada pertenezca a cada una de las clases de salida.

6. Entrenamiento

6.1. Modelo Simple

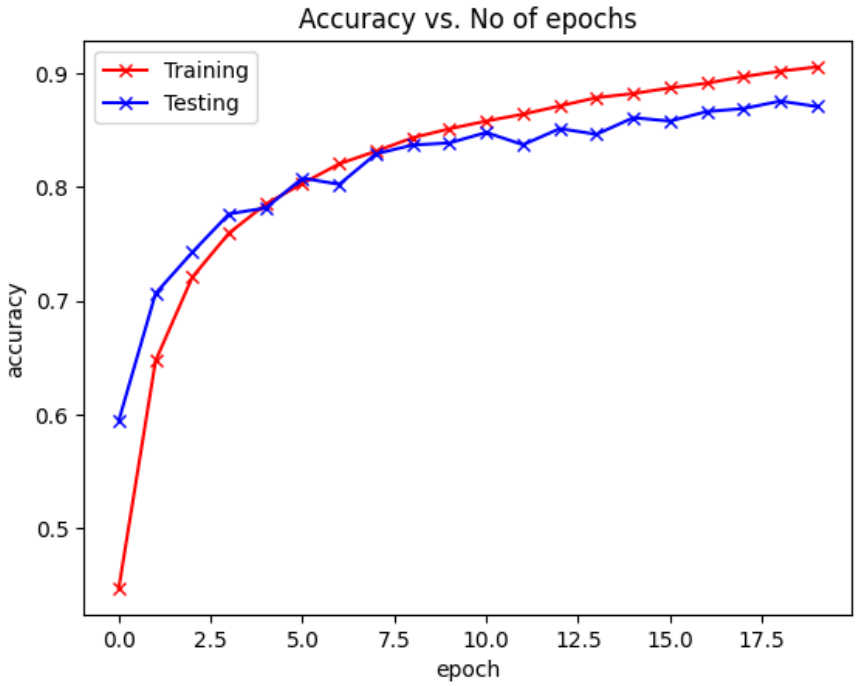

Figura 1: Precisión contra épocas para el Modelo Simple.

Para el conjunto de entrenamiento, la precisión comienza en aproximadamente 0.45 y aumenta gradualmente hasta alcanzar alrededor de 0.91 en la última etapa de entrenamiento. Para el conjunto de prueba, la precisión comienza en aproximadamente 0.59 y aumenta hasta alrededor de 0.87 en la última etapa de prueba. La precisión en el conjunto de prueba es generalmente más bajo que en el conjunto de entrenamiento, indicando un posible sobre ajuste de los datos de entrenamiento y no generalizando tan bien en nuevos datos.

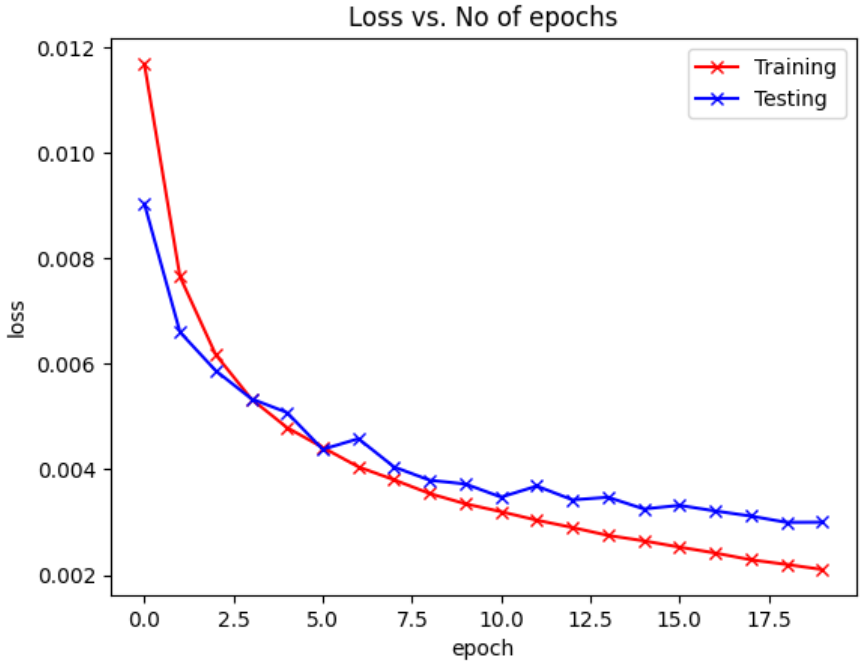

Figura 2: Pérdida contra épocas para el Modelo Simple.

Para el conjunto de entrenamiento, la pérdida promedio disminuye constantemente a medida que avanza el entrenamiento, desde aproximadamente 0.012 hasta 0.002. Para el conjunto de prueba, la pérdida promedio también disminuye a lo largo del tiempo de entrenamiento, desde aproximadamente 0.009 hasta 0.003. El descenso en la pérdida indica que la red neuronal está aprendiendo a realizar predicciones más precisas a medida que se entrena.

En conclusión, a medida que avanzamos en las etapas de entrenamiento, la precisión y la pérdida promedio mejoran tanto en el conjunto de entrenamiento como en el conjunto de prueba, lo que indica que no hay subajuste significativo en los datos proporcionados.Sin embargo, la precisión y la pérdida promedio en el entrenamiento son ligeramente mejores que en el conjunto de prueba, lo cual sugiere un posible sobre ajuste. No obstante, los valores de precisión y pérdida en el conjunto de prueba también mejoran a medida que avanza el entrenamiento, lo que indica que el modelo sigue generalizando bien los datos nuevos.

6.2. Modelo ResNet-18

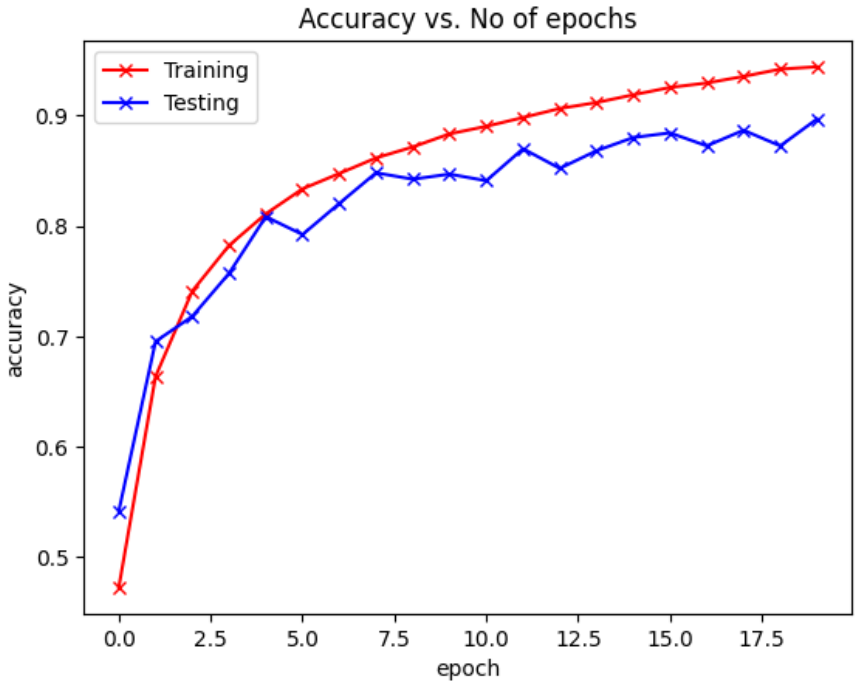

Figura 3: Precisión contra épocas para el Modelo ResNet-18.

Para el conjunto de entrenamiento, la precisión comienza en aproximadamente 0.47 y aumenta gradualmente hasta alcanzar alrededor de 0.94 en la última etapa de entrenamiento. Para el conjunto de prueba, la precisión comienza en aproximadamente 0.54 y aumenta hasta alrededor de 0.90 en la última etapa de prueba. la precisión en el conjunto de prueba es generalmente más bajo que en el conjunto de entrenamiento, indicando un posible sobre ajuste de los datos de entrenamiento y no generalizando tan bien en nuevos datos.

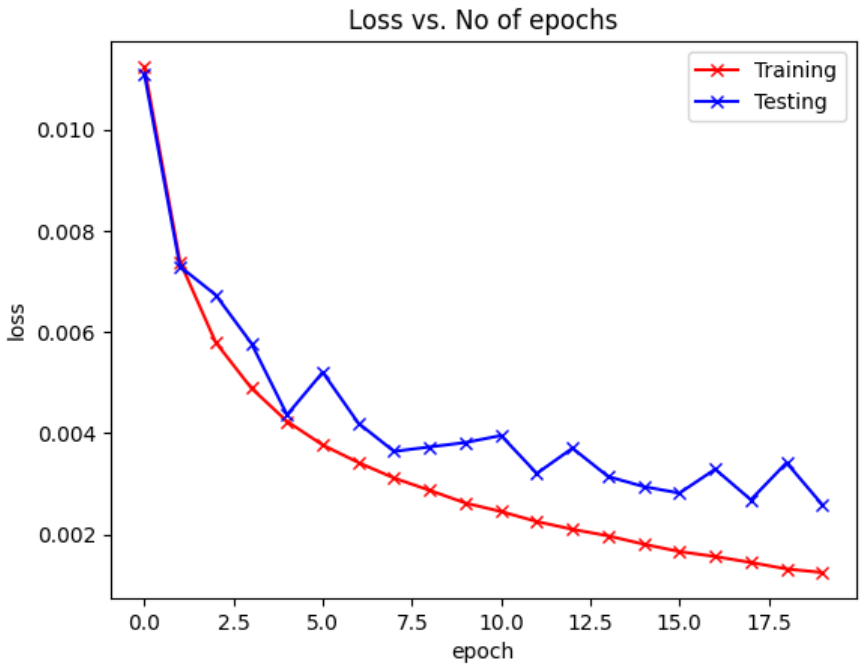

Figura 4: Pérdida contra épocas para el Modelo ResNet-18.

Para el conjunto de entrenamiento, la pérdida promedio disminuye constantemente a medida que avanza el entrenamiento, desde aproximadamente 0.011 hasta 0.001. Para el conjunto de prueba, la pérdida promedio también disminuye a lo largo del tiempo de entrenamiento, desde aproximadamente 0.011 hasta 0.003. El descenso en la pérdida indica que la red neuronal está aprendiendo a realizar predicciones más precisas a medida que se entrena.

A medida que avanzamos en las fases de entrenamiento, se observa una mejora tanto en la precisión como en la pérdida promedio de ambos conjuntos, lo que indica que no hay un subajuste significativo. Sin embargo, se evidencia que la precisión y la pérdida promedio en el entrenamiento son ligeramente superiores a las de prueba, lo que sugiere un posible sobreajuste. No obstante, se observa también un incremento en los valores de precisión y pérdida en el conjunto de prueba a medida que avanza el proceso de adiestramiento, lo que demuestra que el modelo sigue siendo capaz de generalizar adecuadamente con nuevos datos.

7. Evaluación

7.1. Matriz de Confusión y Métricas

7.1.1. Modelo Simple

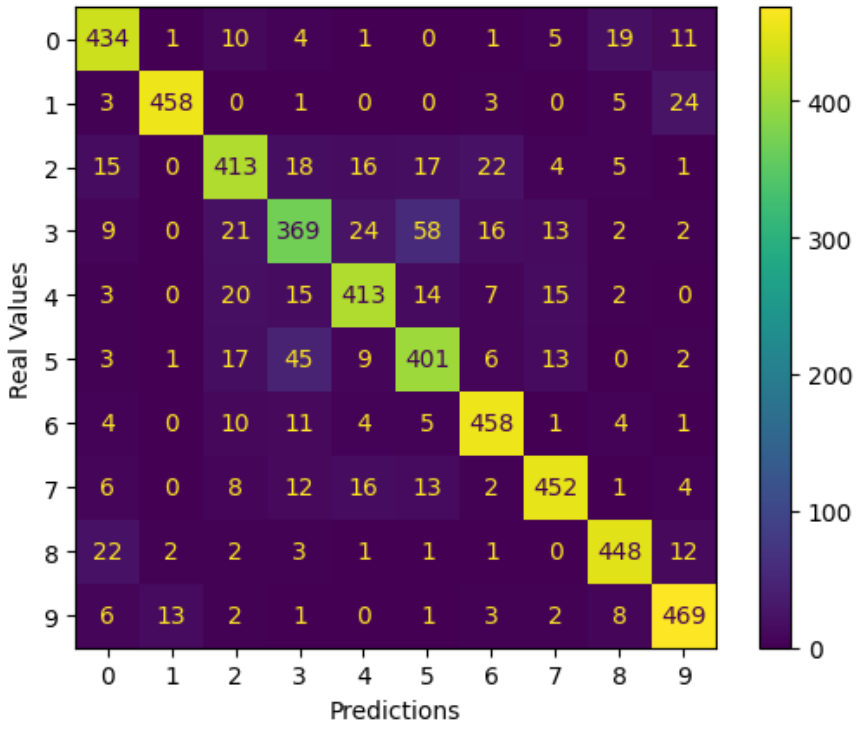

Figura 5: Matriz de Confusión para el Modelo Simple

La matriz de confusión brinda información sobre cómo el modelo clasifica diferentes elementos en distintas clases. Al analizar esta matriz, podemos observar que la diagonal principal, que representa las predicciones correctas, presenta valores altos para las siguientes clases: aviones (0), automóviles (1), ciervos (4), ranas (6), caballos (7), barcos (8) y camiones (9). Estos valores indican que el modelo tiene una buena capacidad de clasificación para estas clases en particular.

Sin embargo, se aprecia que la red neuronal enfrenta ciertas dificultades para clasificar correctamente la clase 2 (pájaros), confundiéndola con las clases 3 (gatos), 4 (ciervos), 5 (perros) y 6 (ranas). Además, en el caso de la clase 3 (gatos), se observan altos errores en la clasificación de la clase 5 (perros) en 58 muestras. De manera similar, la clase 5 (perros) presenta un problema similar, clasificando incorrectamente 45 muestras como clase 3 (gatos).

Por otro lado, contamos con métricas para evaluar el rendimiento de la red neuronal. El modelo ha demostrado un excelente desempeño en datos que no había visto previamente durante el entrenamiento, con una pérdida de validación de 0.00328. Esto indica que el modelo puede realizar predicciones precisas en nuevos datos y generalizar correctamente. Además, la exactitud del modelo fue del 86.3 %, lo que significa que clasificó correctamente el 86.3 % de las muestras en el conjunto de validación.

La precisión del modelo, que mide la precisión de las predicciones positivas, fue de 0.86434, lo que implica que el 86.43 % de las muestras clasificadas como positivas fueron realmente positivas. Asimismo, el recall del modelo fue del 86.3 %, lo que indica que identificó correctamente el 86.3 % de las muestras positivas. En general, el puntaje F1 de 0.8634114385562254 refleja un buen equilibrio entre la precisión y el recall, lo que sugiere un rendimiento sólido en la clasificación de las diferentes clases.

En general, el modelo presenta un rendimiento sólido, con una precisión, recall y puntuación F1 superiores al 86 %. La matriz de confusión muestra una distribución equilibrada de las clasificaciones para cada clase, aunque presenta bastantes errores. Con lo cuál, el modelo ha demostrado un desempeño normal en la clasificación, con métricas sólidas y una confianza razonable en sus predicciones.

7.1.2. Modelo ResNet-18

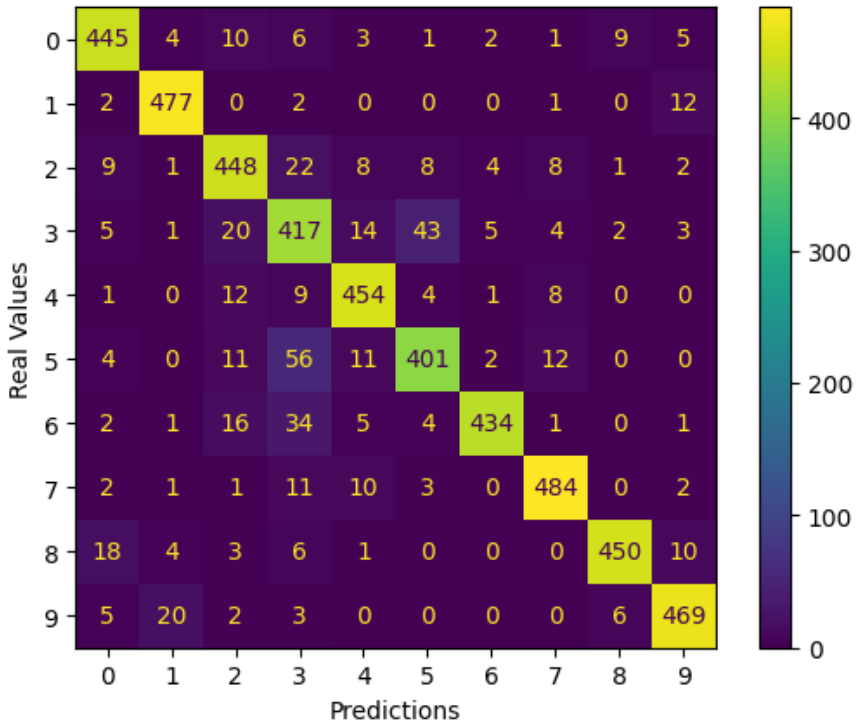

Figura 6: Matriz de Confusión para el Modelo ResNet-18

La matriz de confusión proporciona información sobre cómo el modelo clasifica diferentes elementos en diferentes clases. Al analizar la matriz, se observa que la diagonal principal, que representa las predicciones correctas, presenta valores altos para varias clases, como: aviones (0), automóviles (1), pájaros (2), ciervos (4), caballos (7), barcos (8) y camiones (9). Esto indica que el modelo muestra una buena capacidad de clasificación para estas clases.

Sin embargo, se identifica una dificultad en la red neuronal para clasificar correctamente la clase 3 (gatos), mostrando un rendimiento aceptable, ya que se observan errores en la clasificación de la clase 5 (perros) en 43 muestras. Además, la clase 5 (perros) presenta un problema similar, clasificando incorrectamente 56 muestras como clase 3 (gatos). Asimismo,

se destaca que la clase 6 (ranas) muestra una clasificación errónea de 34 muestras como clase 3 (gatos).

Por otro lado, disponemos de métricas para evaluar el rendimiento de la red neuronal. El modelo ha demostrado un rendimiento excelente en datos que no había visto previamente durante el entrenamiento, con una pérdida de validación de 0.00269. Esto indica que el modelo puede realizar predicciones precisas en nuevos datos y generalizar correctamente. Además, la exactitud del modelo fue del 89.58 %, llo que indica que el 89.58 % de las muestras en el conjunto de validación fueron clasificadas correctamente por la red neuronal.

La precisión del modelo, que mide la precisión de las predicciones positivas, fue de 0.89600, lo que implica que el 89.60 % de las muestras clasificadas como positivas fueron realmente positivas. Asimismo, el recall del modelo fue del 89.58 %, lo que indica que identificó correctamente el 89.58 % de las muestras positivas. En general, el puntaje F1 de 0.8952536718985028 refleja un buen equilibrio entre la precisión y el recall, lo que sugiere un rendimiento sólido en la clasificación de las diferentes clases.

En general, el modelo ha demostrado un rendimiento superior en la clasificación de imágenes, con una precisión, recall y puntuación F1 superiores al 89 %. La matriz de confusión muestra una distribución equilibrada de las clasificaciones para cada clase. Indicando que el modelo ha logrado clasificar las imágenes de manera efectiva.

7.2. Validación Local

7.2.1. Modelo Simple

Figura 7: Validación Local para el Modelo Simple.

De la imagen anterior, se observa que el modelo logra una buena clasificación para la mayoría de las clases de imágenes, como aviones, automóviles, ciervos, caballos, barcos y camiones. Presentando algunas dificultades en la clasificación de otras clases. La clase de pájaro falla en una de tres, confundiéndolo con un gato. La clase de gato falla en dos de seis, confundiendo entre pájaro y perro. La clase perro falla en dos de seis, confundiéndolo con pájaros y venados. La clase rana falla en una de dos, confundiéndolo con pájaros. Estas dificultades mencionadas coinciden con los patrones observados en la matriz de confusión previamente mencionada. A pesar de sus fallas, el modelo logra clasificar correctamente seis de las diez clases, lo cual indica un rendimiento bastante excelente.

7.2.2. Modelo ResNet-18

Figura 8: Validación Local para el Modelo ResNet-18.

De la imagen anterior, se observa que el modelo logra una buena clasificación para la mayoría de las clases de imágenes, como aviones, ciervos, ranas, caballos y camiones. Falla al identificar un auto, identificándolo como rana. Uno de cuatro pájaros fue tomado como perro. Uno de seis gatos fue identificado como pájaro. Tres de seis perros fueron tomados entre gatos y venados. Uno de dos barcos fue tomado como avión. Estas dificultades coinciden parcialmente con los patrones observados en la matriz de confusión previamente mencionada. A pesar de sus fallas, el modelo logra clasificar correctamente cinco de las diez clases, lo cual indica un rendimiento bueno.

8. Conclusiones

Al analizar las métricas, se concluye que ResNet-18 destaca como la opción más potente y efectiva en términos de rendimiento y capacidad de aprendizaje, especialmente en tareas de visión por computadora con conjuntos de datos grandes y complejos. Sin embargo, es importante tener en cuenta que esta ventaja viene acompañada de un desafío significativo: la demanda considerable de recursos computacionales.

Por otro lado, al considerar imágenes externas, se observó que el modelo Sencillo logró una clasificación más precisa y eficiente en comparación con ResNet-18. Esta red demostró una mayor capacidad para identificar patrones y características relevantes en las imágenes, lo que resultó en una mejor precisión en la tarea de clasificación. Además, la simplicidad de esta red facilitó su implementación y entrenamiento, lo que la convierte en una opción atractiva para aplicaciones que requieran una clasificación precisa de imágenes externas.

En general, la elección entre el modelo Sencillo y ResNet-18 depende del contexto y los requisitos específicos del problema. Si el conjunto de datos es pequeño o relativamente simple, y los recursos computacionales y el tiempo de entrenamiento son limitados, una red neuronal sencilla puede ser suficiente. Sin embargo, si el conjunto de datos es grande y complejo, y se busca un rendimiento excepcional en tareas de visión por computadora, ResNet-18 sería una elección más adecuada, a pesar de requerir más recursos y tiempo de entrenamiento.

Referencias

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, DOI 10.1109/CVPR.2009.5206848, pp. 248–255, 06 2009. [Online]. Available: https://www.researchgate.net/publication/221361415_ImageNet_a_Large-Scale_Hierarchical_Image_Database

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[3] A. Esteva, B. Kuprel, R. Novoa, J. Ko, S. Swetter, H. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, DOI 10.1038/nature21056, 01 2017. [Online]. Available: https://www.nature.com/articles/nature21056

[4] M. Jyotiyana, N. Kesswani, and M. Kumar, "A deep learning approach for classification and diagnosis of parkinson's disease," *Research Square*, DOI 10.21203/rs.3.rs-254647/v1, 02 2021. [Online]. Available: https://www.researchgate.net/publication/352455681_A_Deep_Learning_Approach_for_Classification_and_Diagnosis_of_Parkinson%27s_Disease

[5] G. H. Alex Krizhevsky, Vinod Nair, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009. [Online]. Available: https://huggingface.co/datasets/cifar10

[6] Digit eye. [Online]. Available: https://github.com/Oktuvida/digit-eye.git