



**UNIVERSIDAD
DE LA FRONTERA**

UNIVERSIDAD DE LA FRONTERA

FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

Tarea 1 – IMA543

Redes Neuronales Avanzadas de Aprendizaje Profundo

Integrantes: Javier Luengo Abarzúa – Monserrath Álvarez Arriagada

Docente: Benjamin Higuera

Fecha: 6 de mayo de 2025

Introducción

El presente informe tiene objetivo el desarrollo de la Tarea 1 del curso IMA543 – Redes Neuronales Avanzadas de Aprendizaje Profundo. En esta tarea se aborda el análisis, diseño e implementación de arquitecturas de redes neuronales, tanto del tipo Densely Connected Convolutional Networks (DenseNet), como del tipo Deep Residual Networks (ResNet)

A lo largo del documento, se describen los fundamentos teóricos empleados, las metodologías utilizadas y los resultados obtenidos en cada uno de los ejercicios planteados. El objetivo es basándose en los códigos vistos en clases, implementar y entrenar los modelos convolucionales densos —DenseNet y ResNet— utilizando el conjunto de datos **FER2013**.

1. Desarrollo DenseNet

En la implementación de los 3 programas se utilizó la función `extraer_imagenes("FER")` y la función `lr_schedule`, donde la primera carga imágenes de 48x48 píxeles en escala de grises, las organiza en conjuntos de entrenamiento y prueba, normaliza los valores de píxeles dividiéndolos en 255 y convierte las etiquetas a formato one hot (7 clases de emociones). La segunda función implementa una reducción programada del learning rate, donde comienza con 1e-3, para deducir por factores específicos después de las épocas 80, 120, 160 y 180, esto evita el estancamiento del entrenamiento y permite un ajuste mejor.

Los modelos comparten la misma arquitectura tipo **DenseNet** compuesta por una capa de convolución inicial, seguida de cuatro bloques densos, cada uno con 12 capas de tipo cuello de botella, que implican convoluciones **1x1** y **3x3**. Capas de transición entre bloques, que comprimen el número de filtros mediante **compression_factor** específico en cada modelo. Pooling global y una capa densa final de activación **softmax** para la clasificación multiclases.

Se consideran callbacks importantes como **ModelCheckpoint** que guarda el mejor modelo según precisión, **LearningRateScheduler** aplica la programación de tasa de aprendizaje y **ReduceLROnPlateau** que reduce el learning rate si la pérdida se estanca.

Los parámetros utilizados para la realización de los modelos DenseNet son los siguientes:

Parámetro (código)	Valor
<code>batch_size</code>	32
<code>epochs</code>	200
<code>data_augmentation</code>	True
<code>num_classes</code>	7
<code>num_dense_blocks</code>	4
<code>growth_rate</code>	12
<code>depth</code>	100
<code>use_max_pool</code>	False
<code>num_bottleneck_layers</code>	12
<code>num_filters_bef_dense_block</code>	24
<code>optimizer</code>	RMSprop(1e-3)
<code>loss</code>	'categorical_crossentropy'
<code>metrics</code>	['acc']

Cuadro 1: Parámetros comunes utilizados en los tres modelos DenseNet

2. Resultados DenseNet

2.1. DenseNet con tasa de compresión 0.3

Luego de ejecutar el archivo en el servidor Khipu, se obtuvo una **Test accuracy de 0.6828** y unos graficos de exactitud y perdida:

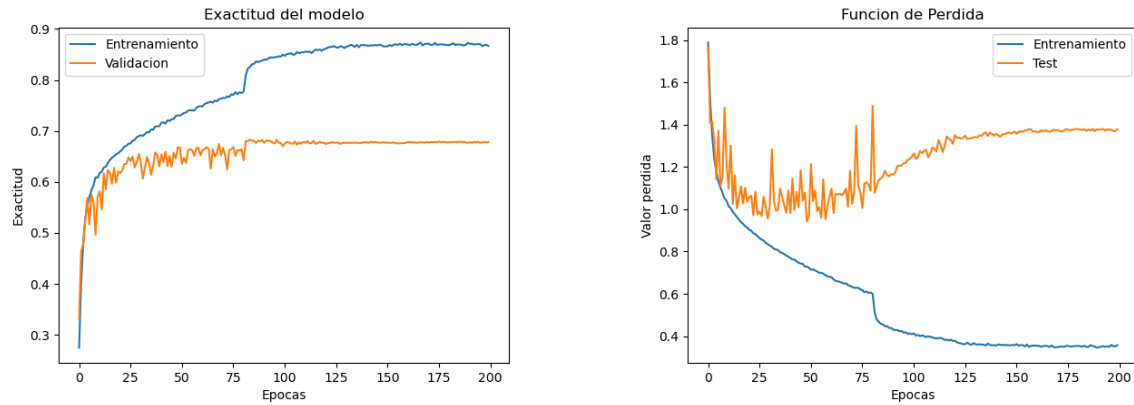


Figura 1: Curvas de exactitud y pérdida para DenseNet con compresión 0.3

2.2. DenseNet con tasa de compresión 0.5

Luego de ejecutar el archivo en el servidor Khipu, se obtuvo **Test accuracy de 0.6909** y unos graficos de exactitud y perdida:

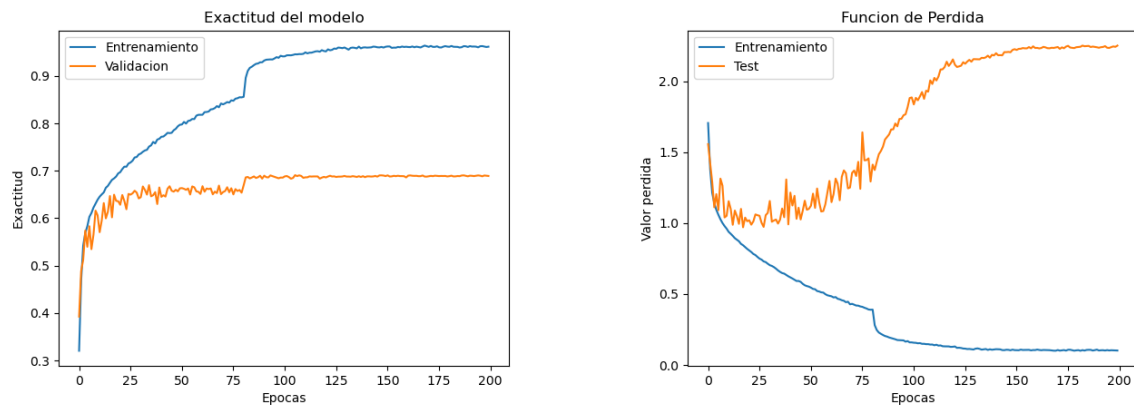


Figura 2: Curvas de exactitud y pérdida para DenseNet con compresión 0.5

2.3. DenseNet con tasa de compresión 0.7

Luego de ejecutar el archivo en el servidor Khipu, se obtuvo **Test accuracy de 0.6934** y unos graficos de exactitud y perdida:

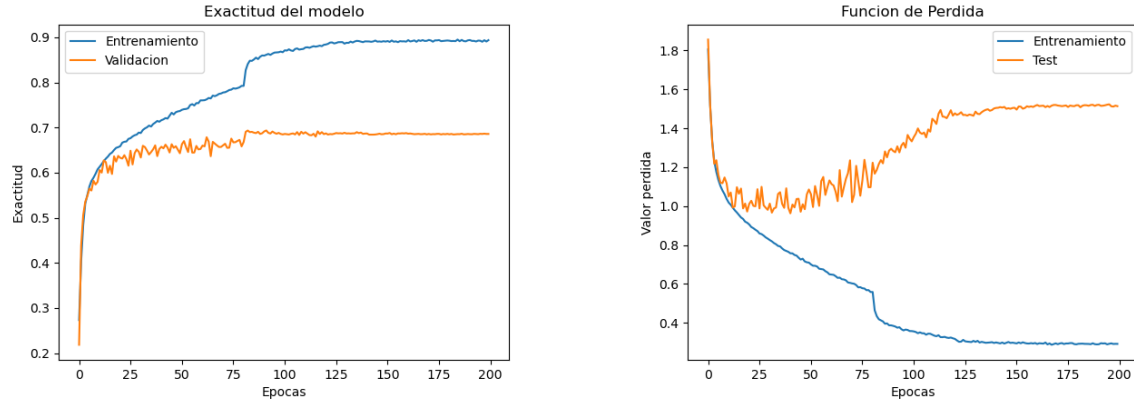


Figura 3: Curvas de exactitud y pérdida para DenseNet con compresión 0.7

3. Conclusiones DenseNet

En la Figura 1, se observa un aumento progresivo en la exactitud en el conjunto de entrenamiento, alcanzando un valor cercano a 0.92 desde aproximadamente la época 80 en adelante, sin embargo, la exactitud en validación se estabiliza en torno a 0.68 sin lograr mejorar significativamente, lo que sugiere un sobreajuste. En el gráfico de pérdida, la curva de entrenamiento disminuye de forma constante hasta llegando a 0.4, mientras que la pérdida en validación se mantiene elevada y con oscilaciones alrededor del 1.2. Esto puede significar que el modelo aprende con eficacia sobre el conjunto de entrenamiento pero no sobre datos no vistos.

En la Figura 2, al igual que el anterior, la exactitud en entrenamiento se eleva hasta aproximadamente 0.93 y se mantiene estable tras la época 100. La exactitud en validación se estanca cerca de 0.65 y no muestra mejoras después de la época 80 aproximadamente. En la pérdida, la curva de entrenamiento sigue una trayectoria descendente hasta llegar aproximadamente a 0.3. Para la pérdida en validación se alcanzan valores superiores a 2, lo que se ve un desajuste aún mayor respecto al modelo anterior, esto sugiere que, aunque la compresión intermedia reduce el número de parámetros, también disminuye la capacidad de generalización del modelo.

Por último, en la Figura 3 se aprecia un factor de compresión 0.7. En este caso, la exactitud en entrenamiento se estabiliza en torno a 0.9 desde la época 100, mientras que la validación se mantiene cerca de 0.65. La pérdida de entrenamiento se reduce progresivamente hasta alcanzar valores cercanos a 0.35, pero la pérdida en validación, si bien no es tan elevada como en el modelo anterior, se estabiliza en un nivel alto, alrededor de 1.5 y esto sugiere que al aumentar el nivel de compresión, el modelo tiene una menor efectividad para capturar patrones complejos en los datos.

En resumen, puede concluirse que los tres modelos comparten un comportamiento similar en cuanto a la capacidad de aprendizaje del entrenamiento, pero se diferencian en la capacidad de desempeño en la validación. El modelo con compresión 0.3 ofrece el mejor rendimiento, aunque con mayor riesgo de sobreajuste. El modelo con compresión 0.5 presenta el peor desempeño en validación, probablemente debido a una pérdida de capacidad significativa. En cambio, el modelo con compresión 0.7 muestra un comportamiento más estable, pero limitado, lo que indica que una compresión excesiva puede comprometer la eficacia del aprendizaje.

4. Desarrollo ResNet

Se utilizó la función `extraer_imagenes()` para cargar las imágenes desde carpetas separadas en entrenamiento y prueba, con 48x48 píxeles en escala de grises, luego se normalizo los valores dividiendo por 255 y codificar las etiquetas en formato **one-hot** para 7 clases. Tanto en ResNet v1 como en v2, se implemento el aprendizaje con `lr_schedule()` tal como para DenseNet.

En ambas implementaciones, la estructura de red neuronal es construida a partir de la función `resnet_layer()`, que define una pila de capas **Conv2D**, **BatchNormalization** y **Activation**, esta función se reutiliza múltiples veces. Las arquitecturas terminan con capas **AveragePooling2D**, **Flatten** y una capa **Dense** con activación **softmax** para clasificación final.

Asimismo, ambos modelos emplean los mismos **callbacks** de entrenamiento: **ModelCheckpoint** para guardar el mejor modelo en validación, **LearningRateScheduler** para aplicar la programación de tasa de aprendizaje, y **ReduceLROnPlateau** para reducir la tasa cuando la función de pérdida de validación se estabiliza.

Los parametros utilizados para la realización de los modelos ResNet son los siguientes:

Parámetro (código)	Valor
<code>batch_size</code>	64
<code>epochs</code>	200
<code>data_augmentation</code>	True
<code>n_clases</code>	7
<code>loss</code>	'categorical_crossentropy'
<code>optimizer</code>	Adam(learning_rate=1e-3)
<code>metrics</code>	['accuracy']
<code>input_shape</code>	(48, 48, 1)

Cuadro 2: Parámetros comunes utilizados en los modelos ResNet v1 y v2

5. Resultados ResNet

5.1. ResNet V1

Luego de ejecutar el archivo en el servidor Khipu, se obtuvo **Test accuracy de 0.6652** y unos graficos de exactitud y perdida:

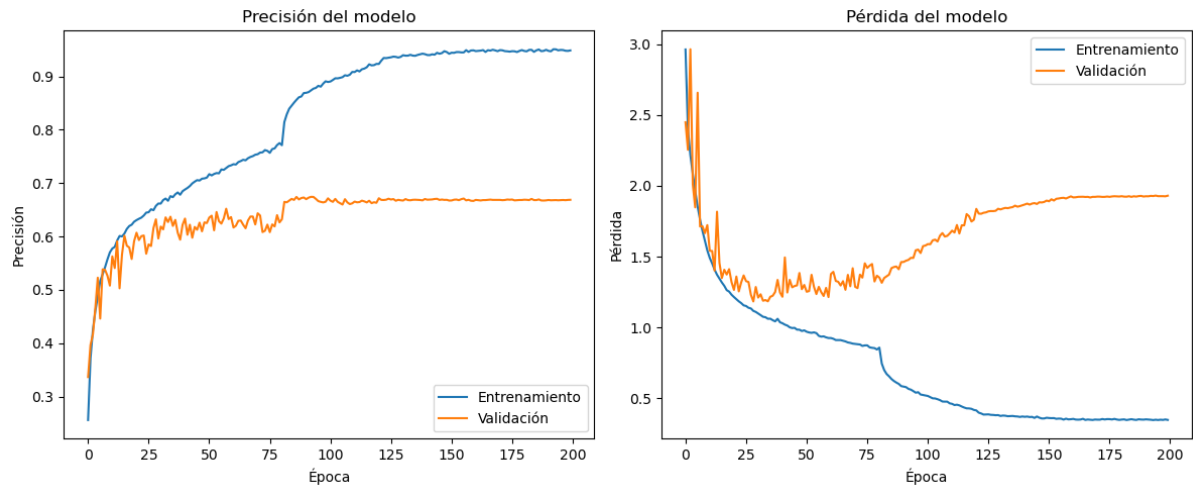


Figura 4: Curvas de exactitud y pérdida para ResNet V1

5.2. ReseNet V2

Luego de ejecutar el archivo en el servidor Khipu, se obtuvo **Test accuracy de 0.6627** y unos graficos de exactitud y perdida:

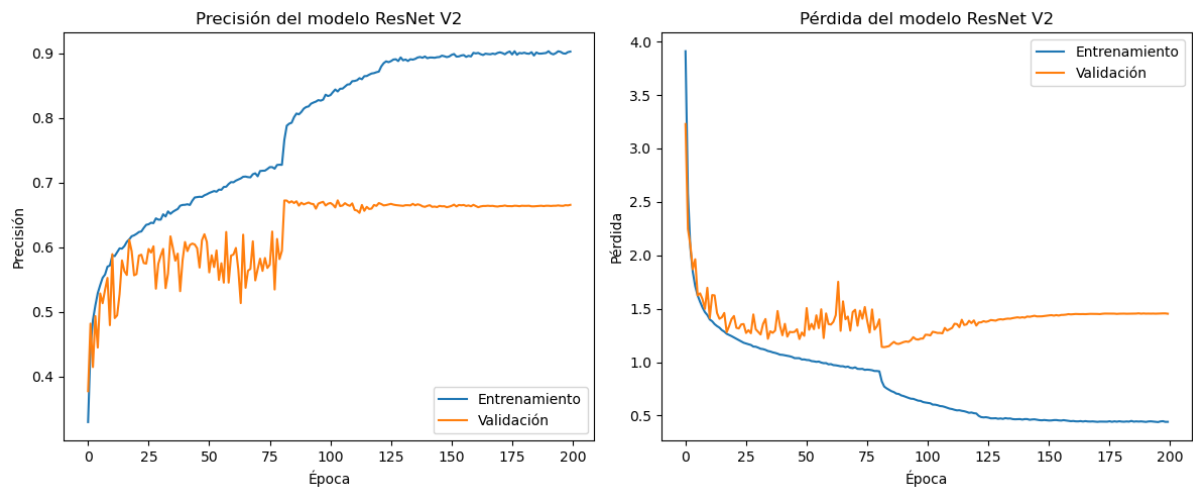


Figura 5: Curvas de exactitud y pérdida para ResNet V1

6. Conclusiones ResNet V1 y V2

En la Figura 1 se observa que la precisión en entrenamiento aumenta progresivamente y alcanza un valor cercano a 0.93 a partir de la época 100, manteniéndose luego estable. La precisión en validación, en cambio, se estabiliza en 0.68 aproximadamente, sin mostrar mejoras relevantes después de la época 100. En cuanto a la pérdida, la curva de entrenamiento disminuye de forma constante hasta un valor cercano a 0.35. Sin embargo, la pérdida de validación presenta oscilaciones considerables desde el inicio y se estabiliza alrededor de 1.9 a partir de la época 120. Esto indica que el modelo ajusta bien los datos de entrenamiento, pero presenta dificultades para los datos de validación.

En la Figura 2, el comportamiento de entrenamiento es similar al modelo anterior, alcanzando una precisión de aproximadamente 0.92 después de la época 100. En validación, la precisión se estabiliza en torno a 0.65. La curva de pérdida en entrenamiento decrece hasta llegar cercanos a 0.35. Por otro lado, la pérdida de validación desciende inicialmente, pero a partir de la época 80 muestra un comportamiento errático con una tendencia a estabilizarse en valores alrededor de 1.5. Esta diferencia sugiere una separación creciente entre el rendimiento en entrenamiento y validación.

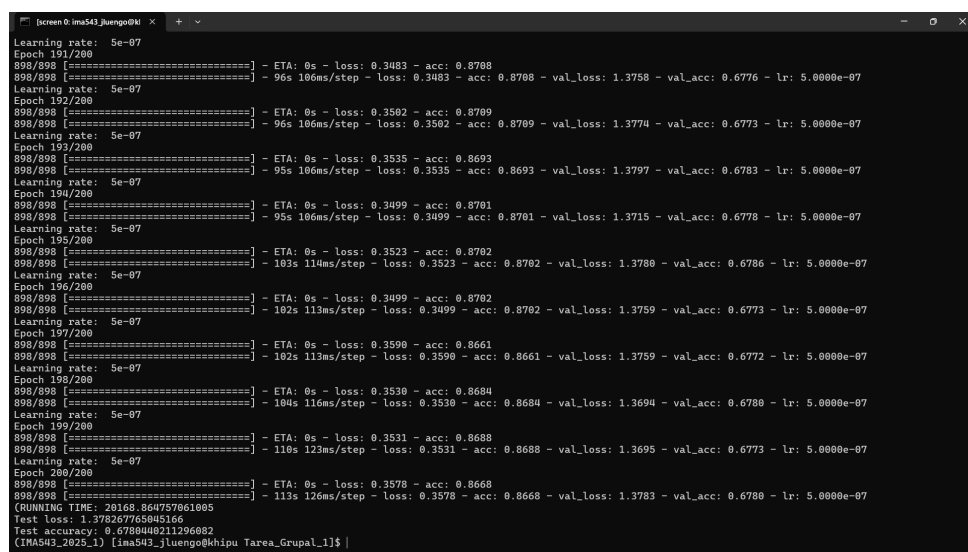
En resumen, aunque ResNet v2 tiene más capas y un diseño distinto, su rendimiento en validación es parecido al ResNet v1. Esto sugiere que aumentar la profundidad y modificar la estructura no garantiza una mejora. En ambos casos, el entrenamiento progresa de manera efectiva, pero los resultados en validación se estabilizan sin mostrar una mejora significativa después de cierta cantidad de épocas.

Anexos

Repositorio Github

https://github.com/LuengoJavier/Tarea_Grupal_RN

Evidencia de ejecución



```
Screen 0: ima543_luengo@k
Learning rate: 5e-07
Epoch 191/200
898/898 [=====] - ETA: 0s - loss: 0.3483 - acc: 0.8788
898/898 [=====] - 96s 106ms/step - loss: 0.3483 - acc: 0.8788 - val_loss: 1.3758 - val_acc: 0.6776 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 192/200
898/898 [=====] - ETA: 0s - loss: 0.3502 - acc: 0.8789
898/898 [=====] - 96s 106ms/step - loss: 0.3502 - acc: 0.8789 - val_loss: 1.3774 - val_acc: 0.6773 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 193/200
898/898 [=====] - ETA: 0s - loss: 0.3535 - acc: 0.8693
898/898 [=====] - 95s 106ms/step - loss: 0.3535 - acc: 0.8693 - val_loss: 1.3797 - val_acc: 0.6783 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 194/200
898/898 [=====] - ETA: 0s - loss: 0.3499 - acc: 0.8701
898/898 [=====] - 95s 106ms/step - loss: 0.3499 - acc: 0.8701 - val_loss: 1.3715 - val_acc: 0.6778 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 195/200
898/898 [=====] - ETA: 0s - loss: 0.3523 - acc: 0.8702
898/898 [=====] - 103s 114ms/step - loss: 0.3523 - acc: 0.8702 - val_loss: 1.3780 - val_acc: 0.6786 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 196/200
898/898 [=====] - ETA: 0s - loss: 0.3499 - acc: 0.8702
898/898 [=====] - 102s 113ms/step - loss: 0.3499 - acc: 0.8702 - val_loss: 1.3759 - val_acc: 0.6773 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 197/200
898/898 [=====] - ETA: 0s - loss: 0.3590 - acc: 0.8661
898/898 [=====] - 102s 113ms/step - loss: 0.3590 - acc: 0.8661 - val_loss: 1.3759 - val_acc: 0.6772 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 198/200
898/898 [=====] - ETA: 0s - loss: 0.3530 - acc: 0.8684
898/898 [=====] - 104s 116ms/step - loss: 0.3530 - acc: 0.8684 - val_loss: 1.3694 - val_acc: 0.6788 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 199/200
898/898 [=====] - ETA: 0s - loss: 0.3531 - acc: 0.8688
898/898 [=====] - 110s 123ms/step - loss: 0.3531 - acc: 0.8688 - val_loss: 1.3695 - val_acc: 0.6773 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 200/200
898/898 [=====] - ETA: 0s - loss: 0.3578 - acc: 0.8668
898/898 [=====] - 113s 126ms/step - loss: 0.3578 - acc: 0.8668 - val_loss: 1.3783 - val_acc: 0.6788 - lr: 5.0000e-07
(RUNNING TIME: 20168.864757861085
Test loss: 1.378267765045166
Test accuracy: 0.6788448211296882
C:\IMA543_2025_1 [ima543_luengo@khipu Tarea_Grupal_1]$
```

Figura 6: Ejecución del modelo DenseNet 0.3.

```
Learning rate: 5e-07
Epoch 191/200
898/898 [=====] - ETA: 0s - loss: 0.1025 - acc: 0.9630
898/898 [=====] - 106s 118ms/step - loss: 0.1025 - acc: 0.9630 - val_loss: 2.2359 - val_acc: 0.6897 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 192/200
898/898 [=====] - ETA: 0s - loss: 0.1040 - acc: 0.9621
898/898 [=====] - 106s 118ms/step - loss: 0.1040 - acc: 0.9621 - val_loss: 2.2398 - val_acc: 0.6895 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 193/200
898/898 [=====] - ETA: 0s - loss: 0.1044 - acc: 0.9622
898/898 [=====] - 106s 118ms/step - loss: 0.1044 - acc: 0.9622 - val_loss: 2.2412 - val_acc: 0.6889 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 194/200
898/898 [=====] - ETA: 0s - loss: 0.1022 - acc: 0.9619
898/898 [=====] - 105s 117ms/step - loss: 0.1022 - acc: 0.9619 - val_loss: 2.2471 - val_acc: 0.6896 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 195/200
898/898 [=====] - ETA: 0s - loss: 0.1051 - acc: 0.9605
898/898 [=====] - 106s 118ms/step - loss: 0.1051 - acc: 0.9605 - val_loss: 2.2360 - val_acc: 0.6903 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 196/200
898/898 [=====] - ETA: 0s - loss: 0.1027 - acc: 0.9628
898/898 [=====] - 105s 117ms/step - loss: 0.1027 - acc: 0.9628 - val_loss: 2.2345 - val_acc: 0.6897 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 197/200
898/898 [=====] - ETA: 0s - loss: 0.1042 - acc: 0.9630
898/898 [=====] - 106s 118ms/step - loss: 0.1042 - acc: 0.9630 - val_loss: 2.2403 - val_acc: 0.6885 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 198/200
898/898 [=====] - ETA: 0s - loss: 0.1033 - acc: 0.9622
898/898 [=====] - 106s 118ms/step - loss: 0.1033 - acc: 0.9622 - val_loss: 2.2451 - val_acc: 0.6902 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 199/200
898/898 [=====] - ETA: 0s - loss: 0.1027 - acc: 0.9612
898/898 [=====] - 106s 118ms/step - loss: 0.1027 - acc: 0.9612 - val_loss: 2.2422 - val_acc: 0.6897 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 200/200
898/898 [=====] - ETA: 0s - loss: 0.1022 - acc: 0.9618
898/898 [=====] - 105s 117ms/step - loss: 0.1022 - acc: 0.9618 - val_loss: 2.2509 - val_acc: 0.6892 - lr: 5.0000e-07
(RUNNING TIME: 21835.962000131607
Test Loss: 2.260896692276081
Test accuracy: 0.6891891956329346
(TMA543_2025_1) [ina543_jluengo@khipu Tarea_Grupal_1]$
```

Figura 7: Ejecución del modelo DenseNet 0.5.

```
Learning rate: 5e-07
Epoch 191/200
898/898 [=====] - ETA: 0s - loss: 0.2895 - acc: 0.8939
898/898 [=====] - 114s 126ms/step - loss: 0.2895 - acc: 0.8939 - val_loss: 1.5212 - val_acc: 0.6853 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 192/200
898/898 [=====] - ETA: 0s - loss: 0.2946 - acc: 0.8928
898/898 [=====] - 117s 130ms/step - loss: 0.2946 - acc: 0.8928 - val_loss: 1.5156 - val_acc: 0.6851 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 193/200
898/898 [=====] - ETA: 0s - loss: 0.2933 - acc: 0.8921
898/898 [=====] - 123s 137ms/step - loss: 0.2933 - acc: 0.8921 - val_loss: 1.5164 - val_acc: 0.6860 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 194/200
898/898 [=====] - ETA: 0s - loss: 0.2940 - acc: 0.8907
898/898 [=====] - 128s 143ms/step - loss: 0.2940 - acc: 0.8907 - val_loss: 1.5185 - val_acc: 0.6856 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 195/200
898/898 [=====] - ETA: 0s - loss: 0.2888 - acc: 0.8927
898/898 [=====] - 127s 141ms/step - loss: 0.2888 - acc: 0.8927 - val_loss: 1.5216 - val_acc: 0.6853 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 196/200
898/898 [=====] - ETA: 0s - loss: 0.2917 - acc: 0.8924
898/898 [=====] - 138s 154ms/step - loss: 0.2917 - acc: 0.8924 - val_loss: 1.5234 - val_acc: 0.6861 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 197/200
898/898 [=====] - ETA: 0s - loss: 0.2938 - acc: 0.8902
898/898 [=====] - 140s 156ms/step - loss: 0.2938 - acc: 0.8902 - val_loss: 1.5135 - val_acc: 0.6858 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 198/200
898/898 [=====] - ETA: 0s - loss: 0.2988 - acc: 0.8936
898/898 [=====] - 138s 152ms/step - loss: 0.2988 - acc: 0.8936 - val_loss: 1.5118 - val_acc: 0.6864 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 199/200
898/898 [=====] - ETA: 0s - loss: 0.2915 - acc: 0.8907
898/898 [=====] - 136s 152ms/step - loss: 0.2915 - acc: 0.8907 - val_loss: 1.5167 - val_acc: 0.6858 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 200/200
898/898 [=====] - ETA: 0s - loss: 0.2912 - acc: 0.8937
898/898 [=====] - 140s 155ms/step - loss: 0.2912 - acc: 0.8937 - val_loss: 1.5141 - val_acc: 0.6858 - lr: 5.0000e-07
(RUNNING TIME: 22126.19722700119
Test Loss: 1.514065140446228
Test accuracy: 0.6850456130796143
(TMA543_2025_1) [ina543_jluengo@khipu Tarea_Grupal_1]$
```

Figura 8: Ejecución del modelo DenseNet 0.7.


```

359/359 [=====] - ETA: 0s - loss: 0.3527 - accuracy: 0.9478
359/359 [=====] - 36s 101ms/step - loss: 0.3527 - accuracy: 0.9478 - val_loss: 1.9269 - val_accuracy: 0.6684 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 192/200
359/359 [=====] - ETA: 0s - loss: 0.3514 - accuracy: 0.9478
359/359 [=====] - 36s 102ms/step - loss: 0.3514 - accuracy: 0.9478 - val_loss: 1.9265 - val_accuracy: 0.6678 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 193/200
359/359 [=====] - ETA: 0s - loss: 0.3500 - accuracy: 0.9509
359/359 [=====] - 36s 101ms/step - loss: 0.3500 - accuracy: 0.9509 - val_loss: 1.9295 - val_accuracy: 0.6680 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 194/200
359/359 [=====] - ETA: 0s - loss: 0.3476 - accuracy: 0.9588
359/359 [=====] - 36s 101ms/step - loss: 0.3476 - accuracy: 0.9588 - val_loss: 1.9282 - val_accuracy: 0.6682 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 195/200
359/359 [=====] - ETA: 0s - loss: 0.3502 - accuracy: 0.9489
359/359 [=====] - 36s 101ms/step - loss: 0.3502 - accuracy: 0.9489 - val_loss: 1.9316 - val_accuracy: 0.6678 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 196/200
359/359 [=====] - ETA: 0s - loss: 0.3480 - accuracy: 0.9496
359/359 [=====] - 36s 102ms/step - loss: 0.3480 - accuracy: 0.9496 - val_loss: 1.9275 - val_accuracy: 0.6680 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 197/200
359/359 [=====] - ETA: 0s - loss: 0.3503 - accuracy: 0.9494
359/359 [=====] - 36s 102ms/step - loss: 0.3503 - accuracy: 0.9494 - val_loss: 1.9278 - val_accuracy: 0.6684 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 198/200
359/359 [=====] - ETA: 0s - loss: 0.3487 - accuracy: 0.9481
359/359 [=====] - 36s 101ms/step - loss: 0.3487 - accuracy: 0.9481 - val_loss: 1.9275 - val_accuracy: 0.6680 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 199/200
359/359 [=====] - ETA: 0s - loss: 0.3520 - accuracy: 0.9478
359/359 [=====] - 36s 101ms/step - loss: 0.3520 - accuracy: 0.9478 - val_loss: 1.9263 - val_accuracy: 0.6685 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 200/200
359/359 [=====] - ETA: 0s - loss: 0.3487 - accuracy: 0.9487
359/359 [=====] - 36s 101ms/step - loss: 0.3487 - accuracy: 0.9487 - val_loss: 1.9307 - val_accuracy: 0.6687 - lr: 5.0000e-07

Tiempo de entrenamiento: 2h 15m 56s
113/113 [=====] - 2s 21ms/step - loss: 1.9426 - accuracy: 0.6651
Test loss: 1.9426
Test accuracy: 0.6651
(IWA543_2025_1) [ina543_jluengo@khipu Tarea_Grupal_1]$ python Punto_b_1.py |

```

Figura 9: Ejecución del modelo ResNet V1.

```

359/359 [=====] - ETA: 0s - loss: 0.4411 - accuracy: 0.9033
359/359 [=====] - 92s 256ms/step - loss: 0.4411 - accuracy: 0.9033 - val_loss: 1.4539 - val_accuracy: 0.6642 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 192/200
359/359 [=====] - ETA: 0s - loss: 0.4452 - accuracy: 0.8999
359/359 [=====] - 92s 256ms/step - loss: 0.4452 - accuracy: 0.8999 - val_loss: 1.4546 - val_accuracy: 0.6638 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 193/200
359/359 [=====] - ETA: 0s - loss: 0.4465 - accuracy: 0.8985
359/359 [=====] - 92s 256ms/step - loss: 0.4465 - accuracy: 0.8985 - val_loss: 1.4539 - val_accuracy: 0.6642 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 194/200
359/359 [=====] - ETA: 0s - loss: 0.4441 - accuracy: 0.9003
359/359 [=====] - 93s 259ms/step - loss: 0.4441 - accuracy: 0.9003 - val_loss: 1.4547 - val_accuracy: 0.6642 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 195/200
359/359 [=====] - ETA: 0s - loss: 0.4413 - accuracy: 0.9033
359/359 [=====] - 92s 257ms/step - loss: 0.4413 - accuracy: 0.9033 - val_loss: 1.4536 - val_accuracy: 0.6647 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 196/200
359/359 [=====] - ETA: 0s - loss: 0.4396 - accuracy: 0.9023
359/359 [=====] - 93s 258ms/step - loss: 0.4396 - accuracy: 0.9023 - val_loss: 1.4546 - val_accuracy: 0.6642 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 197/200
359/359 [=====] - ETA: 0s - loss: 0.4457 - accuracy: 0.9000
359/359 [=====] - 93s 258ms/step - loss: 0.4457 - accuracy: 0.9000 - val_loss: 1.4564 - val_accuracy: 0.6640 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 198/200
359/359 [=====] - ETA: 0s - loss: 0.4477 - accuracy: 0.8996
359/359 [=====] - 92s 257ms/step - loss: 0.4477 - accuracy: 0.8996 - val_loss: 1.4562 - val_accuracy: 0.6650 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 199/200
359/359 [=====] - ETA: 0s - loss: 0.4416 - accuracy: 0.9019
359/359 [=====] - 92s 257ms/step - loss: 0.4416 - accuracy: 0.9019 - val_loss: 1.4563 - val_accuracy: 0.6647 - lr: 5.0000e-07
Learning rate: 5e-07
Epoch 200/200
359/359 [=====] - ETA: 0s - loss: 0.4419 - accuracy: 0.9027
359/359 [=====] - 92s 256ms/step - loss: 0.4419 - accuracy: 0.9027 - val_loss: 1.4535 - val_accuracy: 0.6656 - lr: 5.0000e-07

Tiempo de entrenamiento: 5h 1m 37s
113/113 [=====] - 7s 65ms/step - loss: 1.4614 - accuracy: 0.6626
Test loss: 1.4614
Test accuracy: 0.6626
(IWA543_2025_1) [ina543_jluengo@khipu Tarea_Grupal_1]$ |

```

Figura 10: Ejecución del modelo ResNet V2.

6.1. Tiempos de ejecución

Modelo	Tiempo de Ejecución
DenseNet (compresión 0.3)	5h 11m 37s
DenseNet (compresión 0.5)	2h 15m 56s
DenseNet (compresión 0.7)	5h 36m 9s
ResNet V1 (depth=104)	6h 8m 38s
ResNet V2 (depth=155)	7h 21m 44s

Cuadro 3: Tiempos de entrenamiento para los modelos ejecutados en Khipu