

Indexación, búsqueda y análisis en repositorios multimedia

Práctica 1 - Indexación de imágenes con CNNs

José Manuel Bustos Muñoz

Santiago Martinez De la Riva

Índice

1. Mejoras 1 y 2: Estudio del efecto del parámetro batchsize, y estudio del efecto del parámetro resizing_factor.
2. Mejora 3: Explore el uso de normalización de imágenes como última operación de una transformación compuesta.
3. Mejora 4: Explore y compare distintas estrategias de optimización.
4. Mejora 5: Explore y compare una aplicación progresiva de un factor Learning rate mediante el elemento lr_scheduler.
5. Mejora 6: Aplique técnicas de Data Augmentation para mejorar la calidad de los datos de entrenamiento. Para realizar nuevas transformaciones, se sugiere utilizar la funcionalidad Compose.

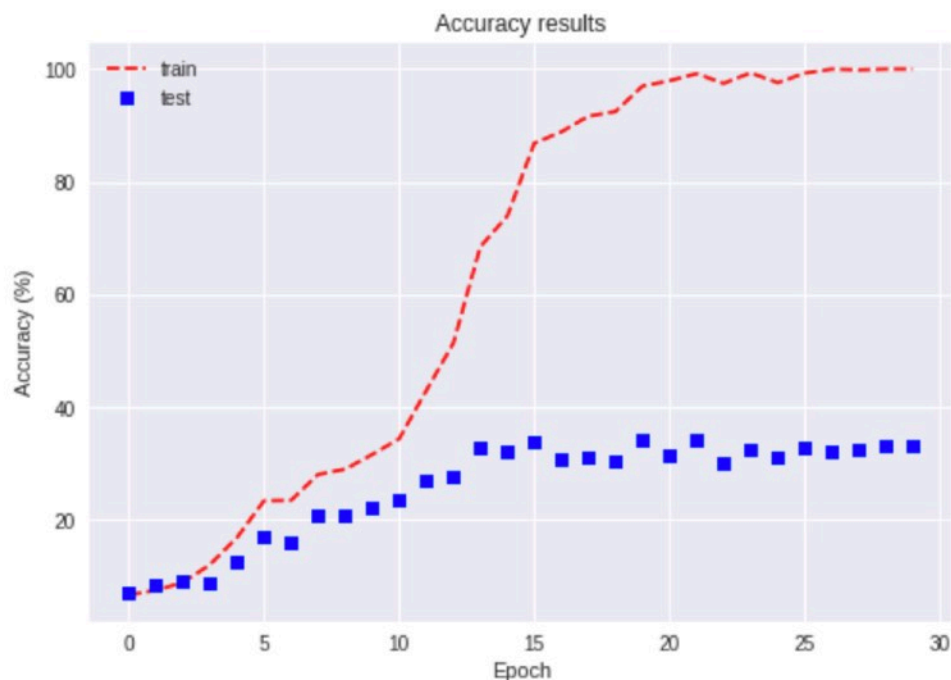
1. Estudio del efecto del parámetro batch_size, y estudio del efecto del parámetro resizing_factor.

Partiendo del punto final de la parte 1 de la práctica, vamos a realizar diferentes ejecuciones modificando el valor del parámetro batch_size, y otros modificando el valor de resizing_factor.

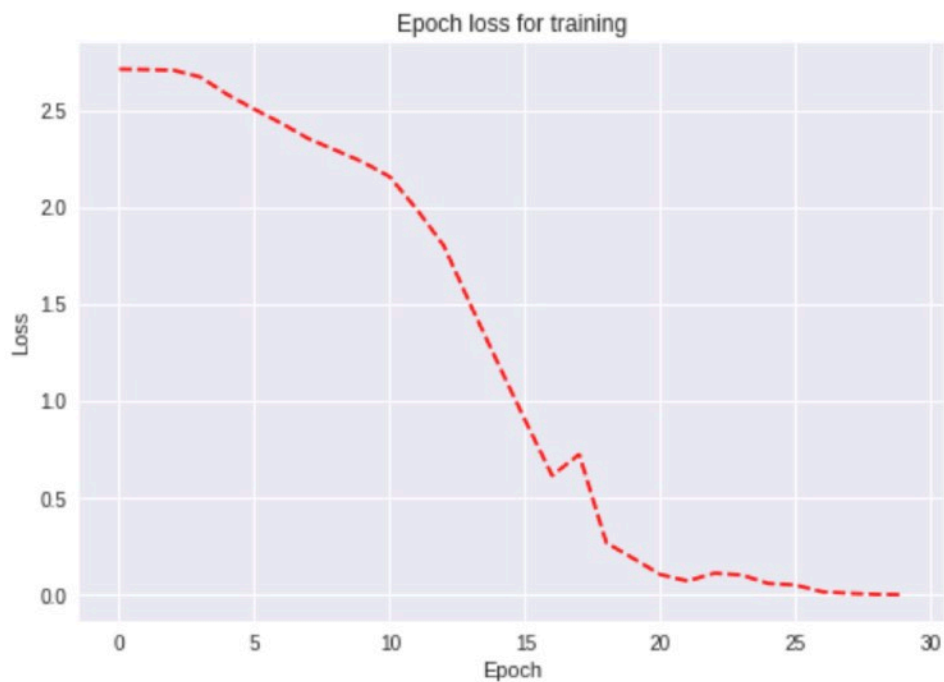
Batch_size es el parámetro que indica el número de ejemplos de entrenamiento en cada epoch. El número óptimo de este parámetro depende del poder de computación del que se disponga, ya que un número mayor requiere mayor capacidad de cómputo, y si no se tiene puede ser perjudicial. Hay que balancear este dato.

Entrenamiento con 30 epochs y batch_size = 4: Como se puede apreciar se llega aproximadamente a un 33% en los casos de test, y existe un claro overfitting. A partir aproximadamente del epoch número 15 ya se mantiene el valor tanto en train como en test.

```
[28, 300] loss: 0.002
Epoch 28, loss: 0.008, accuracy train: 99.867 % (correct 1498 total 1500 ) accuracy test: 32.663 % (correct 0975 total 2985 )
[29, 100] loss: 0.004
[29, 200] loss: 0.003
[29, 300] loss: 0.003
Epoch 29, loss: 0.003, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 33.065 % (correct 0987 total 2985 )
[30, 100] loss: 0.002
[30, 200] loss: 0.001
[30, 300] loss: 0.001
Epoch 30, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 33.300 % (correct 0994 total 2985 )
Finished Training (580.271302938 secs)
```



También podemos ver como baja la pérdida a medida que se ejecutan los epochs, hasta llegar con los datos de training a una pérdida práctica de 0.

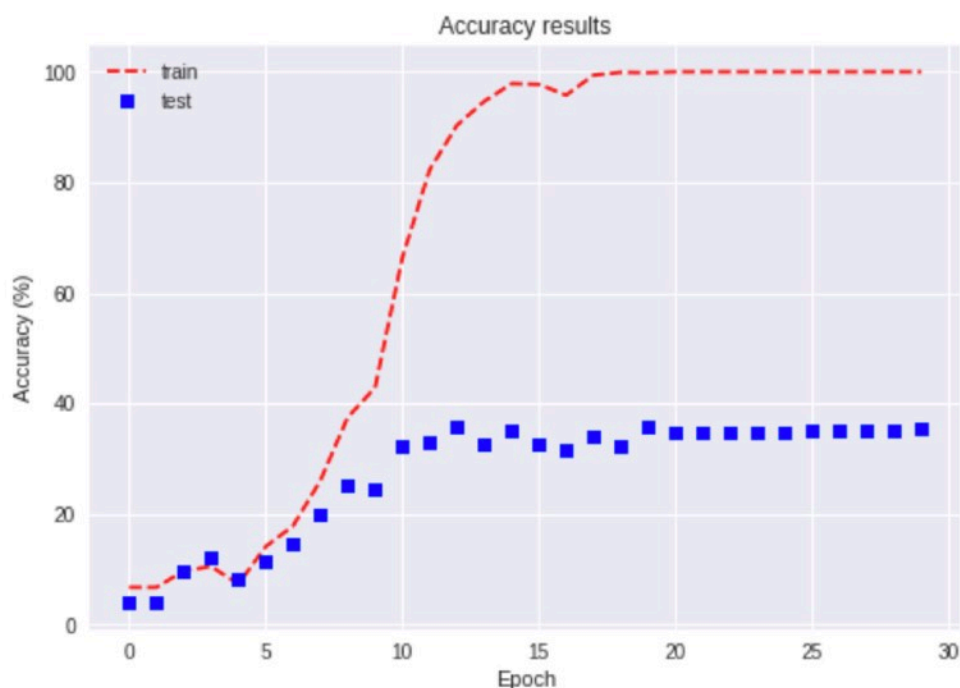


Los resultados obtenidos por clase:

```
Accuracy of Bedroom : 22 %  
Accuracy of Coast : 48 %  
Accuracy of Forest : 31 %  
Accuracy of Highway : 55 %  
Accuracy of Industrial : 14 %  
Accuracy of InsideCity : 16 %  
Accuracy of Kitchen : 11 %  
Accuracy of LivingRoom : 20 %  
Accuracy of Mountain : 23 %  
Accuracy of Office : 28 %  
Accuracy of OpenCountry : 41 %  
Accuracy of Store : 31 %  
Accuracy of Street : 48 %  
Accuracy of Suburb : 52 %  
Accuracy of TallBuilding : 40 %  
Total test data = 2985 images
```

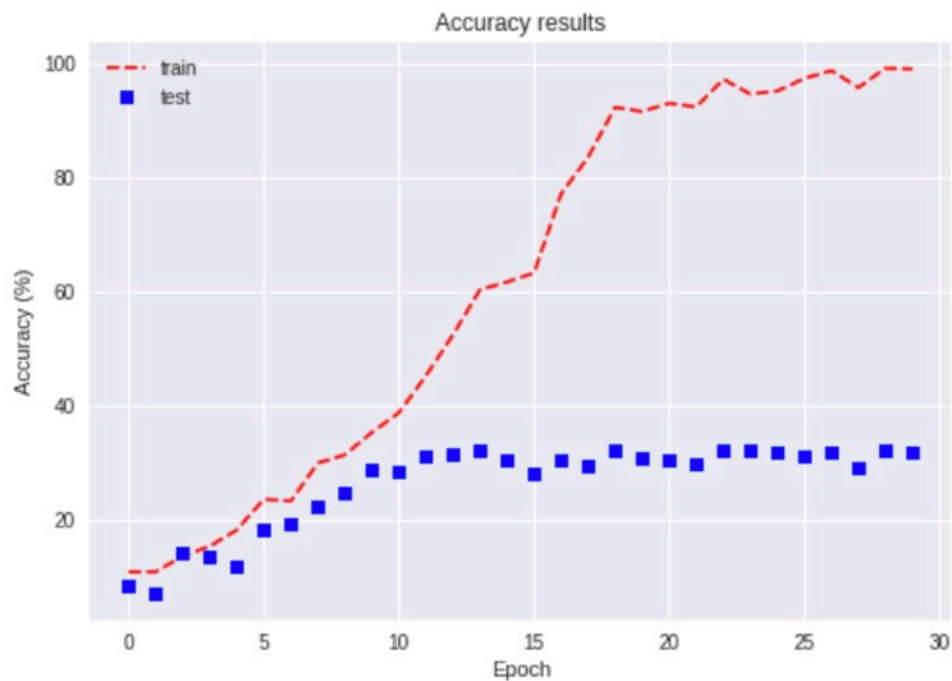
Entrenamiento con 30 epochs y batch_size = 2: Mantenemos el mismo número de epochs, y bajamos el valor de batch_size a 2. Se obtiene una ligera mejoría pero es insignificante. Lo que se puede observar diferente a la ejecución anterior es que parece que al bajar el número del parámetro batch_size respecto a la anterior ejecución se ha producido un aprendizaje más rápido y se ha alcanzado a partir del epoch 10 el estado overfitting anterior donde ya no se crece con los datos de test y se alcanza el 100% con los datos de training.

```
[48, /00] loss: 0.000
Epoch 28, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 35.209 % (correct 1051 total 2985 )
[29, 100] loss: 0.000
[29, 200] loss: 0.000
[29, 300] loss: 0.000
[29, 400] loss: 0.000
[29, 500] loss: 0.000
[29, 600] loss: 0.000
[29, 700] loss: 0.000
Epoch 29, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 35.109 % (correct 1048 total 2985 )
[30, 100] loss: 0.000
[30, 200] loss: 0.000
[30, 300] loss: 0.000
[30, 400] loss: 0.000
[30, 500] loss: 0.000
[30, 600] loss: 0.000
[30, 700] loss: 0.000
Epoch 30, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 35.310 % (correct 1054 total 2985 )
Finished Training (692.411590099 secs)
```



Entrenamiento con 30 épocas y batch_size = 8: aumentando el parámetro de batch_size al contrario que en la anterior ejecución se llega más tarde al 100% en los datos de training, y se llega a un peor resultado en los datos de test, aunque no muy lejano a la primera ejecución.

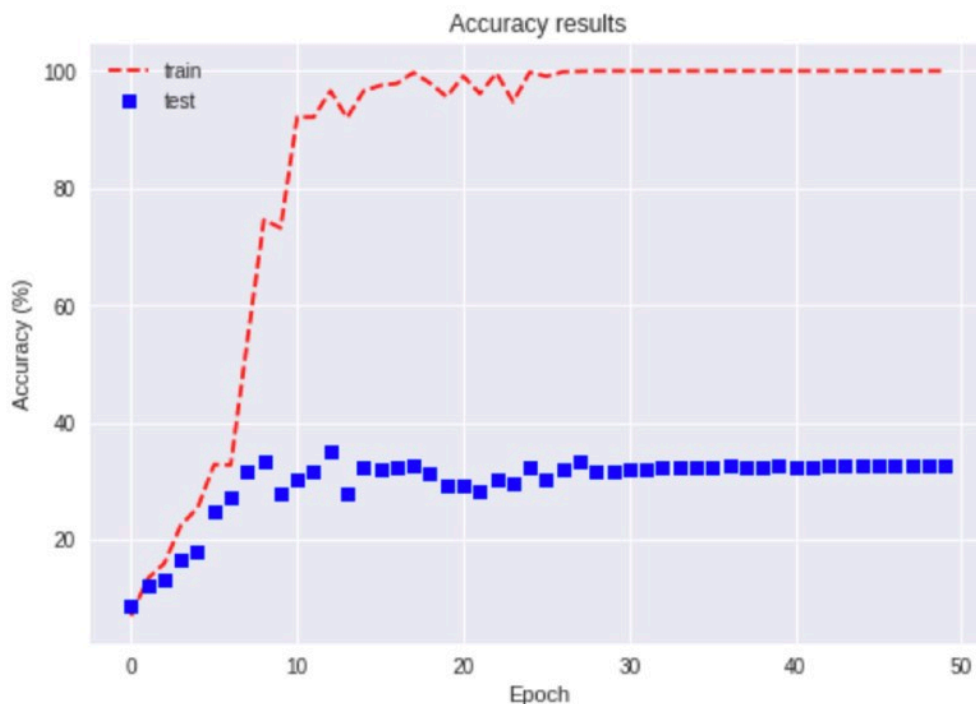
```
[25, 100] loss: 0.234, accuracy train: 95.133 % (correct 1427 total 1500 ) accuracy test: 31.658 % (correct 0945 total 2985 )
[26, 100] loss: 0.154
Epoch 26, loss: 0.130, accuracy train: 97.333 % (correct 1460 total 1500 ) accuracy test: 31.055 % (correct 0927 total 2985 )
[27, 100] loss: 0.161
Epoch 27, loss: 0.156, accuracy train: 98.733 % (correct 1481 total 1500 ) accuracy test: 31.591 % (correct 0943 total 2985 )
[28, 100] loss: 0.096
Epoch 28, loss: 0.132, accuracy train: 95.733 % (correct 1436 total 1500 ) accuracy test: 28.978 % (correct 0865 total 2985 )
[29, 100] loss: 0.071
Epoch 29, loss: 0.058, accuracy train: 99.133 % (correct 1487 total 1500 ) accuracy test: 31.993 % (correct 0955 total 2985 )
[30, 100] loss: 0.067
Epoch 30, loss: 0.053, accuracy train: 99.000 % (correct 1485 total 1500 ) accuracy test: 31.658 % (correct 0945 total 2985 )
Finished Training (599.324135065 secs)
```



Entrenamiento con 50 epochs y batch_size = 2: se hacen dos pruebas con el número de epochs igual a 50, tanto con batch_size = 2, como batch_size = 8. En este caso se obtienen mejores resultados con el valor mayor de batch_size, pero son resultados en test que siempre se mueven entre el 31% y el 38%.

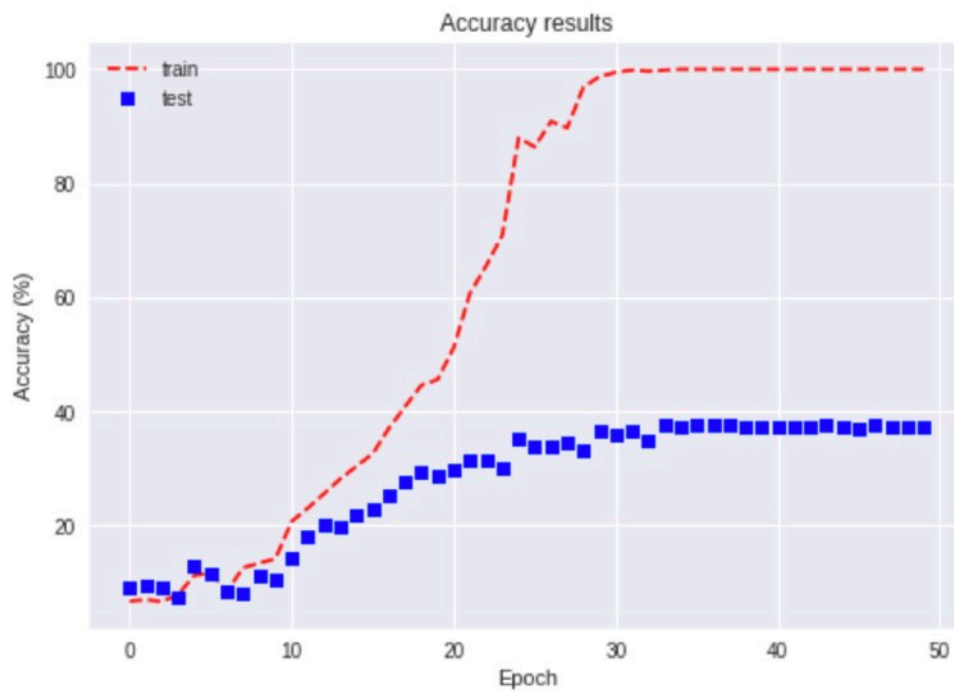
Entendemos que en este caso aumentar de 30 a 50 el número de epochs no debe ser algo significativo, ya que en todas las ejecuciones antes del epoch 30 ya se ha alcanzado el 100% en datos de training y no se crece más con los datos de test.

```
[47, 100] loss: 0.000
Epoch 47, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 32.697 % (correct 0976 total 2985 )
[48, 100] loss: 0.000
[48, 200] loss: 0.000
[48, 300] loss: 0.000
[48, 400] loss: 0.000
[48, 500] loss: 0.000
[48, 600] loss: 0.000
[48, 700] loss: 0.000
Epoch 48, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 32.697 % (correct 0976 total 2985 )
[49, 100] loss: 0.000
[49, 200] loss: 0.000
[49, 300] loss: 0.000
[49, 400] loss: 0.000
[49, 500] loss: 0.000
[49, 600] loss: 0.000
[49, 700] loss: 0.000
Epoch 49, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 32.663 % (correct 0975 total 2985 )
[50, 100] loss: 0.000
[50, 200] loss: 0.000
[50, 300] loss: 0.000
[50, 400] loss: 0.000
[50, 500] loss: 0.000
[50, 600] loss: 0.000
[50, 700] loss: 0.000
Epoch 50, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 32.697 % (correct 0976 total 2985 )
Finished Training (1178.77136183 secs)
```



Entrenamiento con 50 epochs y batch_size = 8:

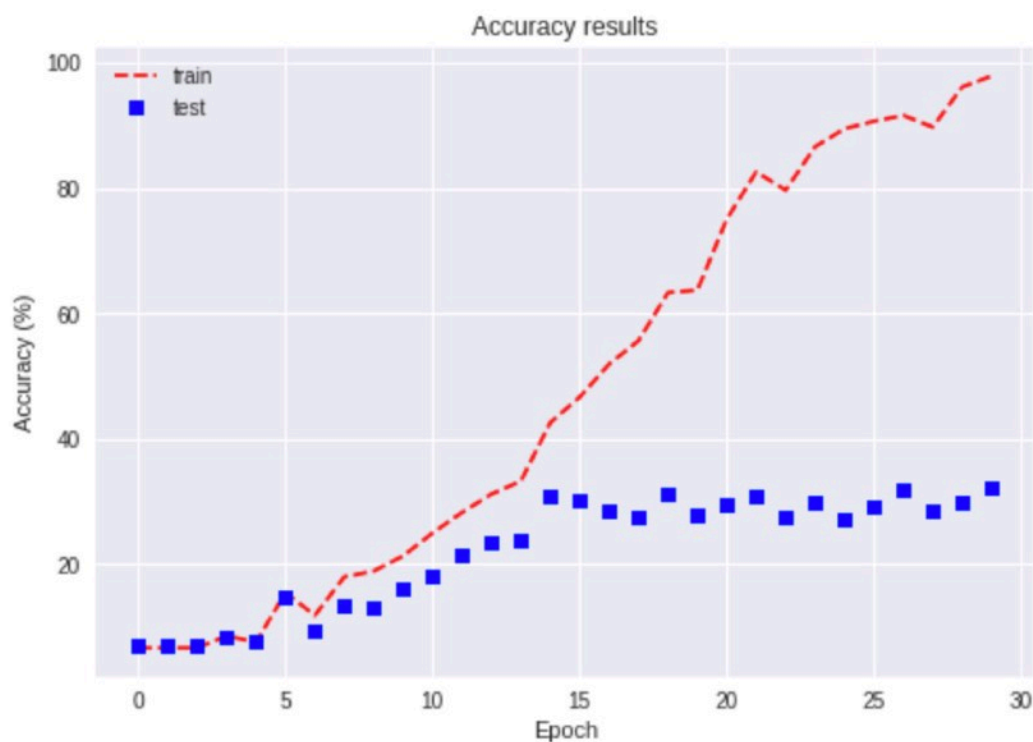
```
[45, 100] loss: 0.001
Epoch 45, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.253 % (correct 1112 total 2985 )
[46, 100] loss: 0.001
Epoch 46, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.119 % (correct 1108 total 2985 )
[47, 100] loss: 0.001
Epoch 47, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.487 % (correct 1119 total 2985 )
[48, 100] loss: 0.001
Epoch 48, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.353 % (correct 1115 total 2985 )
[49, 100] loss: 0.001
Epoch 49, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.219 % (correct 1111 total 2985 )
[50, 100] loss: 0.001
Epoch 50, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.454 % (correct 1118 total 2985 )
Finished Training (959.139224052 secs)
```



Entrenamiento con `resizing_factor = 64x64`: ahora ejecutamos modificando el parámetro `resizing_factor`. Hacemos 3 pruebas, con valores de 64x64, 256x256 y 512x512. El número de epochs para las 3 ejecuciones será de 30, y el parámetro `batch_size` lo ejecutamos con valor de 4, que serían los valores de partida.

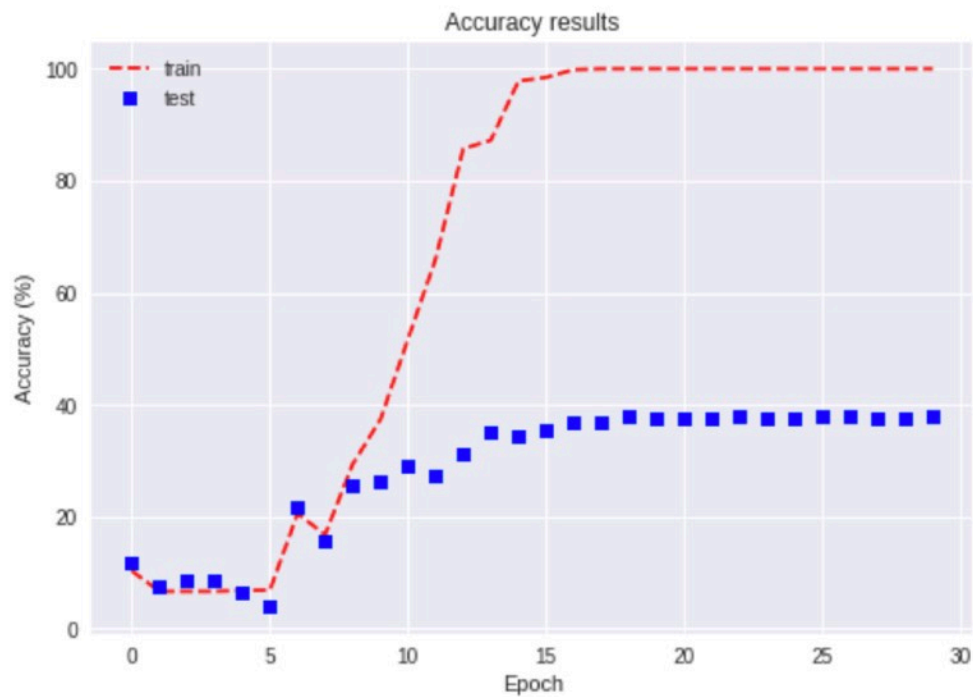
En la ejecución con un valor de `resizing_factor` menor al de partida, se obtienen ligeramente peores resultados, y al aumentar el valor de parámetro se obtiene una cierta mejoría con el punto de partida. Los resultados son mejores con el valor de 256x256, con lo que podemos suponer que dicho aumento viene bien a la hora de entrenar.

```
[25, 300] loss: 0.443
Epoch 25, loss: 0.455, accuracy train: 89.467 % (correct 1342 total 1500 ) accuracy test: 27.303 % (correct 0815 total 2985 )
[26, 100] loss: 0.486
[26, 200] loss: 0.324
[26, 300] loss: 0.392
Epoch 26, loss: 0.398, accuracy train: 90.667 % (correct 1360 total 1500 ) accuracy test: 29.112 % (correct 0869 total 2985 )
[27, 100] loss: 0.282
[27, 200] loss: 0.340
[27, 300] loss: 0.256
Epoch 27, loss: 0.302, accuracy train: 91.600 % (correct 1374 total 1500 ) accuracy test: 31.826 % (correct 0950 total 2985 )
[28, 100] loss: 0.233
[28, 200] loss: 0.322
[28, 300] loss: 0.313
Epoch 28, loss: 0.320, accuracy train: 89.733 % (correct 1346 total 1500 ) accuracy test: 28.710 % (correct 0857 total 2985 )
[29, 100] loss: 0.259
[29, 200] loss: 0.248
[29, 300] loss: 0.221
Epoch 29, loss: 0.241, accuracy train: 96.133 % (correct 1442 total 1500 ) accuracy test: 30.050 % (correct 0897 total 2985 )
[30, 100] loss: 0.169
[30, 200] loss: 0.131
[30, 300] loss: 0.219
Epoch 30, loss: 0.166, accuracy train: 97.867 % (correct 1468 total 1500 ) accuracy test: 32.094 % (correct 0958 total 2985 )
Finished Training (530.109058857 secs)
```



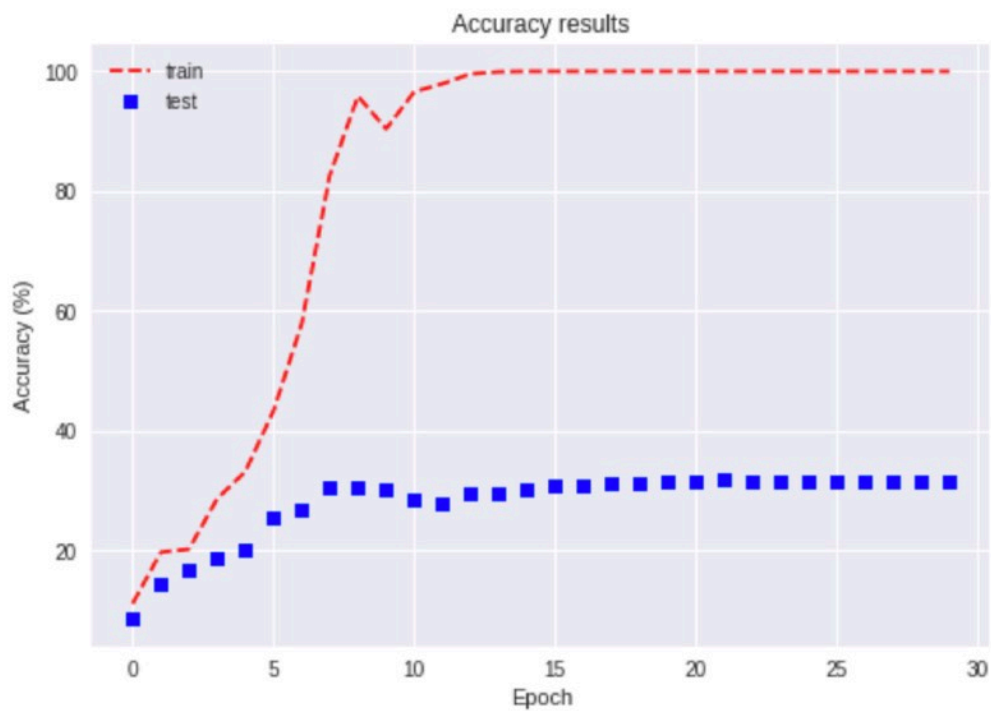
Entrenamiento con `resizing_factor = 256x256`: los mejores resultados obtenidos, cercanos al 38%.

```
[40, 300] loss: 0.000
Epoch 26, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.789 % (correct 1128 total 2985 )
[27, 100] loss: 0.000
[27, 200] loss: 0.000
[27, 300] loss: 0.000
Epoch 27, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.923 % (correct 1132 total 2985 )
[28, 100] loss: 0.000
[28, 200] loss: 0.000
[28, 300] loss: 0.000
Epoch 28, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.688 % (correct 1125 total 2985 )
[29, 100] loss: 0.000
[29, 200] loss: 0.000
[29, 300] loss: 0.000
Epoch 29, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.755 % (correct 1127 total 2985 )
[30, 100] loss: 0.000
[30, 200] loss: 0.000
[30, 300] loss: 0.000
Epoch 30, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 37.789 % (correct 1128 total 2985 )
Finished Training (788.87548995 secs)
```



Entrenamiento con resizing_factor = 512x512:

```
[26, 300] loss: 0.000
Epoch 26, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 31.591 % (correct 0943 total 2985 )
[27, 100] loss: 0.000
[27, 200] loss: 0.000
[27, 300] loss: 0.000
Epoch 27, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 31.591 % (correct 0943 total 2985 )
[28, 100] loss: 0.000
[28, 200] loss: 0.000
[28, 300] loss: 0.000
Epoch 28, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 31.558 % (correct 0942 total 2985 )
[29, 100] loss: 0.000
[29, 200] loss: 0.000
[29, 300] loss: 0.000
Epoch 29, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 31.591 % (correct 0943 total 2985 )
[30, 100] loss: 0.000
[30, 200] loss: 0.000
[30, 300] loss: 0.000
Epoch 30, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 31.591 % (correct 0943 total 2985 )
Finished Training (2380.38449192 secs)
```



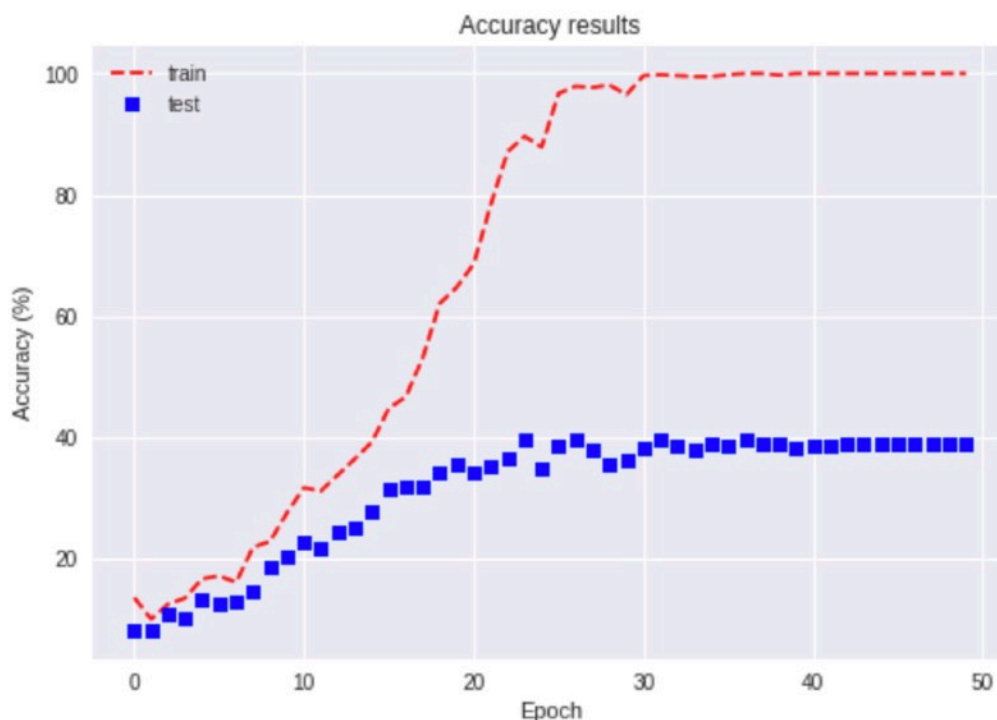
Ejecución con ciertos parámetros después de las pruebas cambiando valores de batch_size y resizing_factor: ahora hacemos una ejecución cambiando tanto el valor de batch_size inicial, como el de resizing_factor, buscando una combinación que nos haga mejorar los resultados obtenidos.

Probamos con batch_size = 16, y resizing_factor = 256x256. Obtenemos prácticamente un 40% en los datos de test, que sería la mejor marca obtenida, aunque como ya hemos visto no son mejoras significativas.

Al aumentar el número de batch_size creemos que debe de mejorar el aprendizaje, pero hay que encontrar el número óptimo de este parámetro según el poder computacional que tengamos, ya que a mayor valor requiere una mayor carga de cómputo y si no lo tenemos sería contraproducente.

Entendemos que el dataset es pequeño y por tanto es complicado hacer un entrenamiento que obtengas buenos resultados y no caiga en el overfitting, además que no estamos ejecutando una red lo suficientemente profunda, que también necesitaría unos datos de mayor volumen para sacarle partido real.

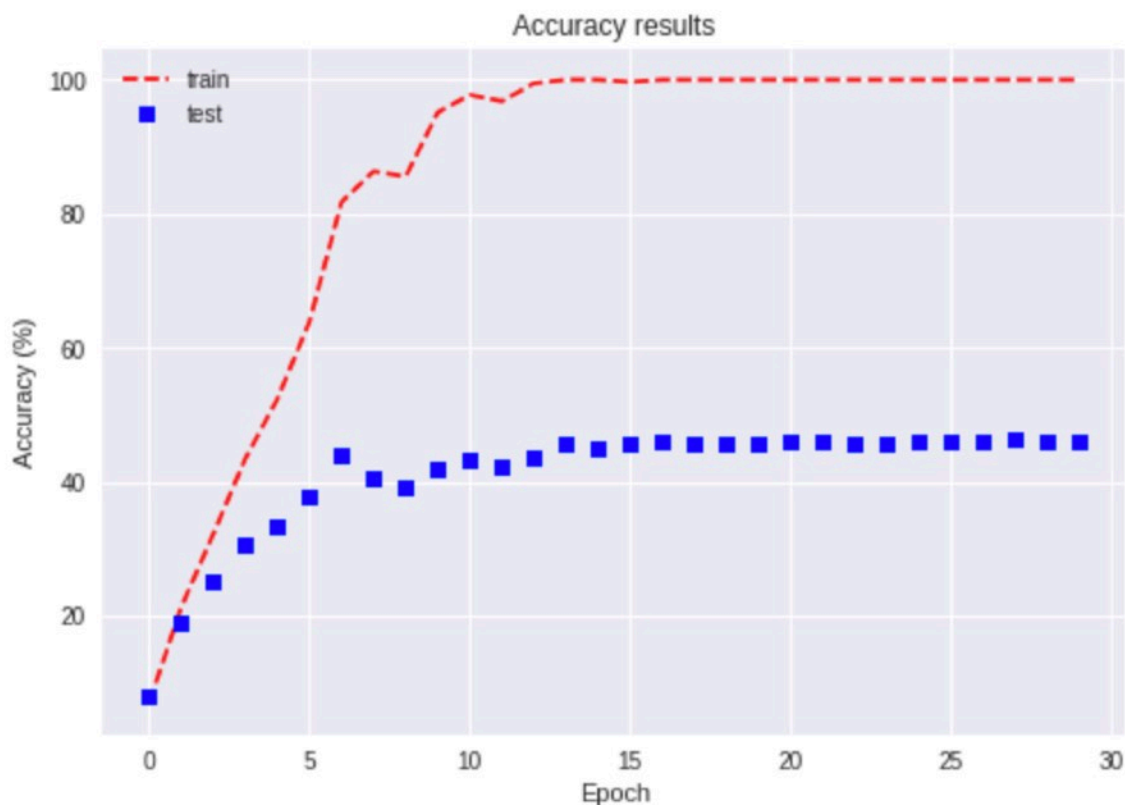
```
Epoch 39, loss: 0.006, accuracy train: 99.733 % (correct 1496 total 1500 ) , accuracy test: 38.827 % (correct 1159 total 2985 )
Epoch 40, loss: 0.004, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.291 % (correct 1143 total 2985 )
Epoch 41, loss: 0.003, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.593 % (correct 1152 total 2985 )
Epoch 42, loss: 0.003, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.693 % (correct 1155 total 2985 )
Epoch 43, loss: 0.002, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.861 % (correct 1160 total 2985 )
Epoch 44, loss: 0.002, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.995 % (correct 1164 total 2985 )
Epoch 45, loss: 0.002, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.861 % (correct 1160 total 2985 )
Epoch 46, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.961 % (correct 1163 total 2985 )
Epoch 47, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.827 % (correct 1159 total 2985 )
Epoch 48, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.827 % (correct 1159 total 2985 )
Epoch 49, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.827 % (correct 1159 total 2985 )
Epoch 50, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) , accuracy test: 38.760 % (correct 1157 total 2985 )
Finished Training (7654.39588809 secs)
```



2. Mejora 3: Explore el uso de normalización de imágenes como última operación de una transformación compuesta.

Se introduce el uso de “transforms.Normalize” al realizar transformaciones en el dataset, y al normalizar las imágenes se consiguen los mejores resultados obtenidos hasta ahora con el dataset “Scenes15”. Se llega aproximadamente al 46% en los datos de test, además se ve como al normalizar se acelera el entrenamiento y obtención de los resultados finales.

```
Epoch 23, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 45.729 % (correct 1365 total 2985 )
Epoch 24, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 45.729 % (correct 1365 total 2985 )
Epoch 25, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 45.997 % (correct 1373 total 2985 )
Epoch 26, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 46.064 % (correct 1375 total 2985 )
Epoch 27, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 45.963 % (correct 1372 total 2985 )
Epoch 28, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 46.198 % (correct 1379 total 2985 )
Epoch 29, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 45.963 % (correct 1372 total 2985 )
Epoch 30, loss: 0.001, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 45.930 % (correct 1371 total 2985 )
Finished Training (4736.26353288 secs)
```

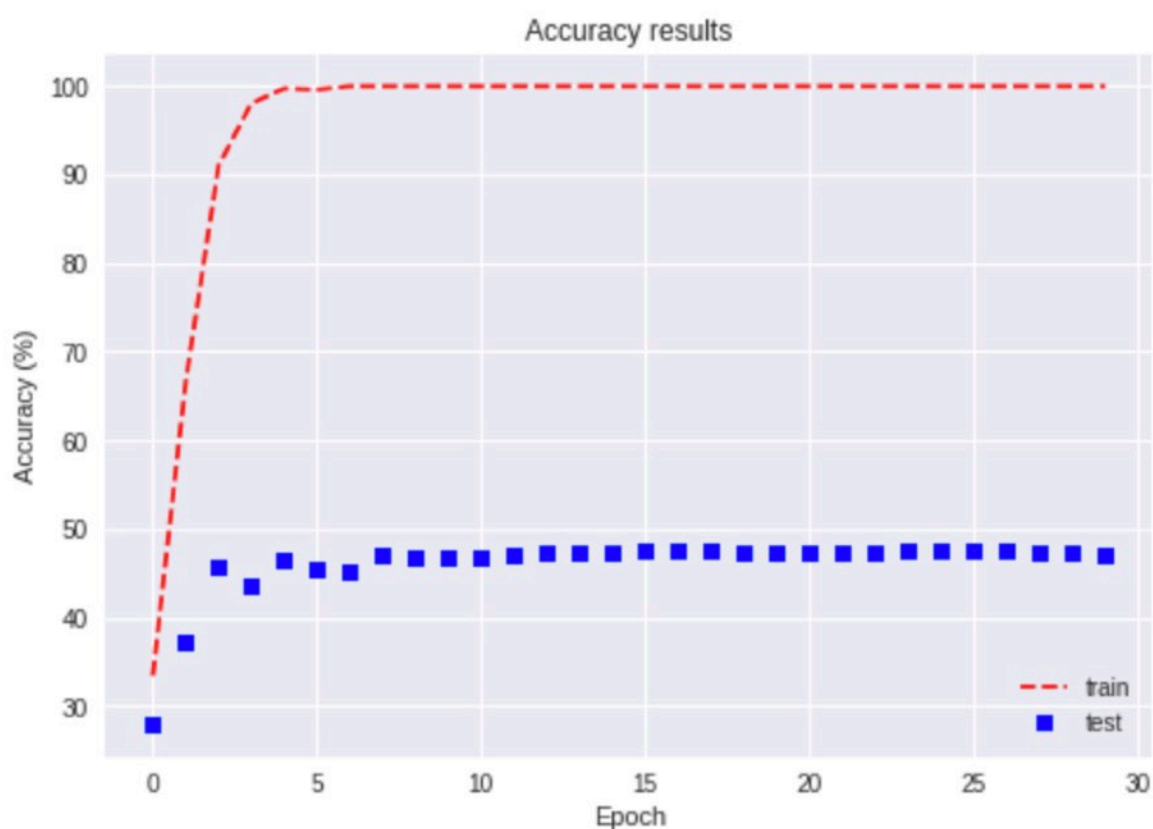


3. Mejora 4: Explore y compare distintas estrategias de optimización.

Después del estudio tras normalizar, se hacen dos ejecuciones más cambiando la estrategia de optimización. Se van a usar dos tipos diferentes de optimizador: Adam y Adadelata.

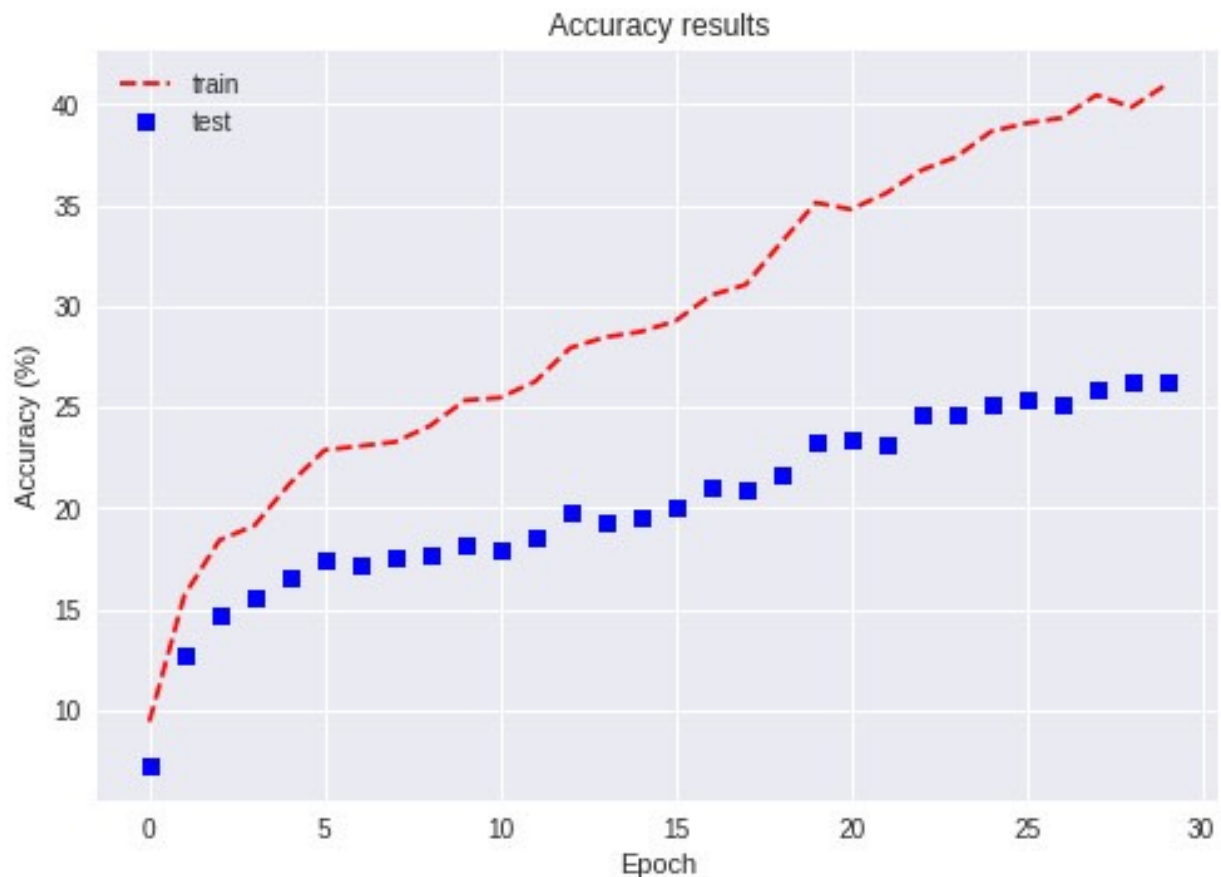
Al probar con el optimizador tipo Adam se ha mejorado algo el resultado obtenido en el anterior paso, hasta llegar al 47%, aunque no es muy significativo. Lo que sí se ha comprobado es que se alcanza antes el resultado tope, y que cada iteración ha tardado mayor tiempo que usando el optimizador anterior. Se puede suponer que con este optimizador se consumen más recursos.

```
Epoch 23, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.337 % (correct 1413 total 2985 )
Epoch 24, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.437 % (correct 1416 total 2985 )
Epoch 25, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.605 % (correct 1421 total 2985 )
Epoch 26, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.638 % (correct 1422 total 2985 )
Epoch 27, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.437 % (correct 1416 total 2985 )
Epoch 28, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.370 % (correct 1414 total 2985 )
Epoch 29, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.337 % (correct 1413 total 2985 )
Epoch 30, loss: 0.000, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 47.035 % (correct 1404 total 2985 )
Finished Training (5285.39624 secs)
```



Luego usando el optimizador tipo Adadelata se utiliza con un pequeño valor del parámetro weight_decay, pero se consiguen valores muy bajos tanto en training como en test. No parece haber un gran overfitting, pero el aprendizaje es muy lento. Habría que realizar pruebas con otro tipo de scheduler o variando algún parámetro, y seguramente con una mayor cantidad de datos para lograr un aprendizaje mayor sin llegar al overfitting.

```
Epoch 21, loss: 2.265, accuracy train: 34.800 % (correct 0522 total 1500 ) accuracy test: 23.384 % (correct 0698 total 2985 )
Epoch 22, loss: 2.242, accuracy train: 35.600 % (correct 0534 total 1500 ) accuracy test: 23.116 % (correct 0690 total 2985 )
Epoch 23, loss: 2.218, accuracy train: 36.733 % (correct 0551 total 1500 ) accuracy test: 24.623 % (correct 0735 total 2985 )
Epoch 24, loss: 2.195, accuracy train: 37.400 % (correct 0561 total 1500 ) accuracy test: 24.690 % (correct 0737 total 2985 )
Epoch 25, loss: 2.172, accuracy train: 38.667 % (correct 0580 total 1500 ) accuracy test: 25.193 % (correct 0752 total 2985 )
Epoch 26, loss: 2.151, accuracy train: 39.067 % (correct 0586 total 1500 ) accuracy test: 25.360 % (correct 0757 total 2985 )
Epoch 27, loss: 2.130, accuracy train: 39.333 % (correct 0590 total 1500 ) accuracy test: 25.159 % (correct 0751 total 2985 )
Epoch 28, loss: 2.110, accuracy train: 40.467 % (correct 0607 total 1500 ) accuracy test: 25.930 % (correct 0774 total 2985 )
Epoch 29, loss: 2.089, accuracy train: 39.867 % (correct 0598 total 1500 ) accuracy test: 26.231 % (correct 0783 total 2985 )
Epoch 30, loss: 2.070, accuracy train: 41.000 % (correct 0615 total 1500 ) accuracy test: 26.231 % (correct 0783 total 2985 )
Finished Training (5068.35163808 secs)
```

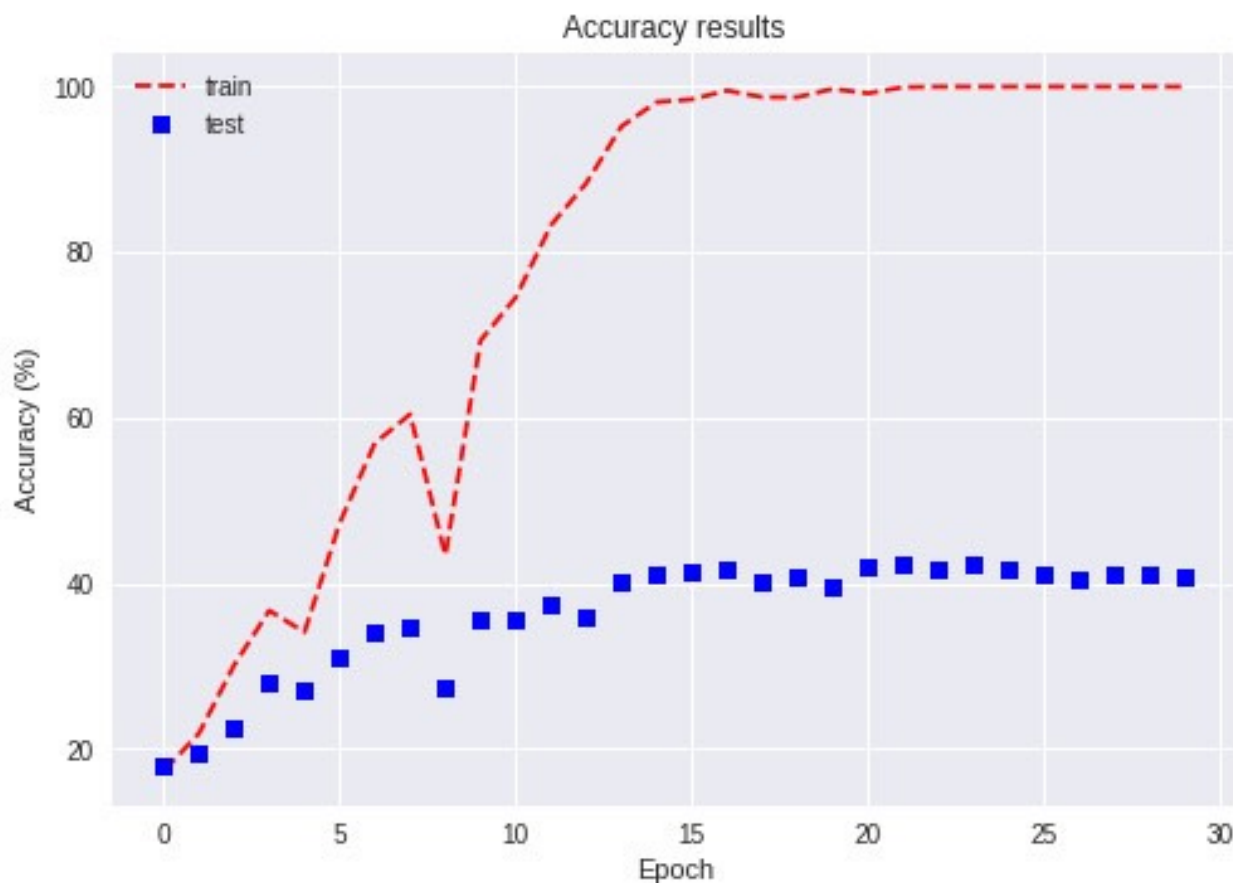


4. Mejora 5: Explore y compare una aplicación progresiva de un factor Learning rate mediante el elemento lr_scheduler.

Se introduce un elemento scheduler tras la optimización. Usamos en este ejemplo el lr_scheduler.CosineAnnealingLR, con 10 pasos.

Se llega al 100% en training, pero en test apenas pasa del 40% así que no llega a los mejores resultados obtenidos en pruebas anteriores.

```
Epoch 25, loss: 0.006, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 41.809 % (correct 1248 total 2985 )
Epoch 26, loss: 0.004, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 41.206 % (correct 1230 total 2985 )
Epoch 27, loss: 0.004, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 40.570 % (correct 1211 total 2985 )
Epoch 28, loss: 0.003, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 40.972 % (correct 1223 total 2985 )
Epoch 29, loss: 0.003, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 41.106 % (correct 1227 total 2985 )
Epoch 30, loss: 0.002, accuracy train: 100.000 % (correct 1500 total 1500 ) accuracy test: 40.704 % (correct 1215 total 2985 )
Finished Training (4897.21188903 secs)
```



5. Mejora 6: Aplique técnicas de Data Augmentation para mejorar la calidad de los datos de entrenamiento. Para realizar nuevas transformaciones, se sugiere utilizar la funcionalidad Compose.

Se han introducido varias sentencias adicionales al realizar el `transforms.compose` a los datos, además de la normalización ya aplicada anteriormente se añaden otras transformaciones como `transforms.RandomRotation` o `transforms.RandomResizedCrop`.

No obtenemos unos buenos resultados, el entrenamiento va obteniendo un aprendizaje bastante lento y parece que con los datos de test se van calcando los datos obtenidos en el training.

```
Epoch 23, loss: 2.078, accuracy train: 30.867 % (correct 0463 total 1500 ) accuracy test: 31.792 % (correct 0949 total 2985 )
Epoch 24, loss: 2.096, accuracy train: 34.600 % (correct 0519 total 1500 ) accuracy test: 31.692 % (correct 0946 total 2985 )
Epoch 25, loss: 2.053, accuracy train: 34.200 % (correct 0513 total 1500 ) accuracy test: 32.362 % (correct 0966 total 2985 )
Epoch 26, loss: 1.996, accuracy train: 27.733 % (correct 0416 total 1500 ) accuracy test: 26.332 % (correct 0786 total 2985 )
Epoch 27, loss: 2.074, accuracy train: 32.600 % (correct 0489 total 1500 ) accuracy test: 32.094 % (correct 0958 total 2985 )
Epoch 28, loss: 2.014, accuracy train: 34.000 % (correct 0510 total 1500 ) accuracy test: 33.233 % (correct 0992 total 2985 )
Epoch 29, loss: 2.022, accuracy train: 35.067 % (correct 0526 total 1500 ) accuracy test: 32.931 % (correct 0983 total 2985 )
Epoch 30, loss: 2.035, accuracy train: 33.333 % (correct 0500 total 1500 ) accuracy test: 31.558 % (correct 0942 total 2985 )
Finished Training (3630.53721809 secs)
```

