Clasificación de insectos basada en imágenes

Raúl García Crespo

Enero 2021

1 Introducción y objetivo

Los insectos representan más del 80% de las diferentes especies de vida animal en el planeta (entre 6 y 10 millones de especies distintas en total), lo que les convierte en la forma de vida más común de la Tierra. La capacidad de distinguir e identificar una sola especie de insecto de entre esta astronómica cifra solo esta al alcance de expertos en la materia y sin embargo podría tener aplicaciones prácticas en ámbitos como por ejemplo la agricultura, para la detección de especies dañinas para los cultivos; o en la protección del medio ambiente, detectando la presencia de especies invasoras procedentes de otros ecosistemas.

En este trabajo se ha empleado el corpus disponible en este enlace, el cual contiene más de 60000 imágenes de insectos repartidos entre 291 especies diferentes, cada una identificada por su número gbif. Las imágenes fueron extraídas a partir de la colección de insectos de la familia Carabidae (comúnmente conocidos como "Ground Beetle") del Museo de Historia Natural de Londres. Debe hacerse notar que dichas imágenes representan especímenes disecados y sobre un fondo neutro, por lo que podrían existir problemas al aplicar los resultados de este trabajo sobre imágenes de especímenes vivos tomadas en su entorno natural, sin embargo la correcta identificación de un número tan elevado de especies, algunas notablemente similares entre sí al pertenecer a la misma familia, ya representa por si mismo un avance significativo del estudio planteado en este trabajo.

Por tanto, el objetivo de este trabajo se basa en la obtención de un clasificador en forma de redes neuronal que logre clasificar con el mayor porcentaje de acierto posible las imágenes del dataset descrito. Para ello será necesario en primer lugar un pequeño preprocesamiento de las imágenes para garantizar su utilidad y que permita una exploración de diferentes estructuras de redes neuronales aplicadas a dichas imágenes.

2 Preproceso

Las imágenes del dataset están estructuradas en carpetas, donde el nombre de cada carpeta es el identificador único de especie del insecto que aparece en las imágenes de dicha carpeta. La inmensa mayoría de las más de 60000 imágenes

de encuentran orientadas verticalmente , es decir, su alto suele ser mayor que su alto, y el insecto que aparece en la imagen sigue esta representación, con el cuerpo colocado en vertical y la cabeza apuntando hacia arriba. Sin embargo, de forma aislada podemos encontrar imágenes con orientación horizontal e incluso imágenes erróneas que no representan ningún insecto, por lo que será necesario rotar o eliminar aquellas imágenes que lo requieran para normalizar todo lo posible el conjunto de imágenes.

Una vez realizado este paso todavía sigue existiendo un problema de normalización, y es que no todas las imágenes poseen las mismas dimensiones, por lo que en el momento de introducirlas como entrada a nuestra red todas las imágenes serán reescaladas a una dimensión de 32x64 píxeles para intentar mantener la relación de aspecto de las imágenes rectangulares todo lo posible y evitar así deformar la imagen.

Figure 1: Ejemplo de imagen mal orientada

3 Experimentación

Una vez se dispone del conjunto total de datos normalizados, este se ha separado en dos conjuntos, uno de test y otro de validación, empleando las funcionalidades de la clase ImageDataGenerator de Keras con un 20% de las imágenes destinadas a validación. Idealmente se debería disponer de un tercer conjunto de test, pero con el objetivo de aprovechar todo lo posible las muestras disponibles, los resultados obtenidos se interpretarán sobre el conjunto de validación. Otras modificaciones que se han aplicado sobre las imagenes de entrenamiento y validación con ImageDataGenerator son la posibilidad de aplicar un pequeño desplazamiento vertical y horizontal, y la posibilidad de invertir horizontalmente las imágenes. Estas especificaciones, junto con un bacth size de 512 y 50 epochs, se mantendrán durante los diferentes experimentos a no ser que se especifique lo contrario, con el propósito de obtener resultados de precisión comparables.

Los experimentos realizados emplean diferentes arquitecturas neuronales a fin de encontrar aquella que arroje mejores resultados sobre nuestro dataset,

algunas ya implementadas en Keras y otros definidas manualmente. Para poder comparar los resultados de las distintas arquitecturas se van a emplear como métricas tanto el porcentaje de acierto de la mejor predicción de la red para cada imagen, como el acierto de las 3 mejores predicciones, siempre sobre el conjunto de validación.

3.1 Arquitectura VGG19

La arquitectura VGG19 es una variante del modelo VGG con una profundidad de 19 capas, orientada a la clasificación de imágenes en gran escala, que en 2014 consiguió la victoria en el desafío de ImageNet (cuyo objetivo es la clasificación de un gran dataset de imágenes repartidas en 1000 clases distintas) con un error inferior al 25% para una sola predicción. Esta arquitectura ya se encuentra implementada en Keras y basta con invocar a la clase VGG19 con nuestro tamaño de imágenes de entrada e inicializando los pesos de la red aleatoriamente para comenzar el entrenamiento. Debido a la lentitud en su aprendizaje, el número de epochs usado para este experimento fue de 100, al final de los cuales los mejores resultados fueron:

- Top-1 prediction: 48.08% de acierto.
- Top-3 prediction: 69.98% de acierto.

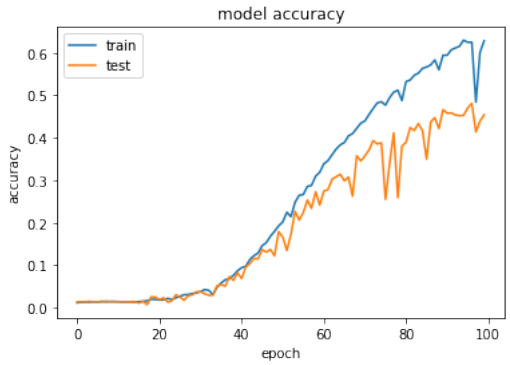

Figure 2: Evolución del aprendizaje de VGG19

Puede observar se en la gráfica de la imagen que durante los 100 epochs del aprendizaje ni el valor de precisión del conjunto de entrenamiento, ni el de validación llegaron a estabilizarse, lo que sugiere que de haber continuado el entrenamiento podrían haberse alcanzado mejores resultados. La razón por la que no se realizó un segundo aprendizaje con mayor número de epochs han sido las limitaciones que presenta el entorno Google Colab, donde se realizaron todos los experimentos de este trabajo. Este entorno solo permite un tiempo limite para el uso de GPU y finaliza automáticamente aquellas sesiones que lo

superen; dado que el entrenamiento para alcanzar estos resultados ya superaba las 7 horas, aumentar la carga de trabajo habría alcanzado los limites impuestos por Colab.

3.2 Arquitectura DenseNet201

DenseNet201, la versión más compleja de las distintas redes convolucionales densamente conectadas que se incluyen entre las arquitecturas ya implementadas de Keras, es la arquitectura que se ha empleado para obtener los siguientes resultados:

- Top-1 prediction: 71.82% de acierto.
- Top-3 prediction: 87.55% de acierto.

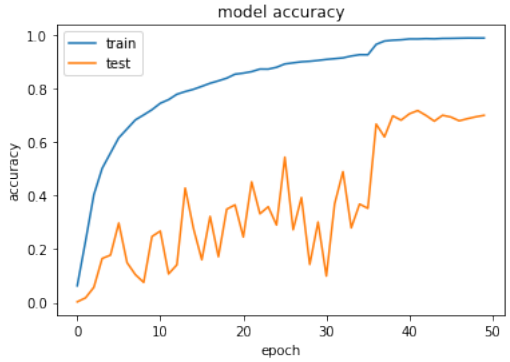

Figure 3: Evolución del aprendizaje de DenseNet 201

A diferencia de la gráfica obtenida para VGG19, la evolución del aprendizaje de DenseNet si que parece indicar una estabilización de la precisión sobre el conjunto de entrenamiento en las últimas epochs. Sin embargo, la evolución del conjunto de validación es mucho más caótica hasta el epoch 30, alternando entre valores dispares y sin indicar que se esté produciendo un aprendizaje, por lo que los resultados obtenidos solo parecen ser fiables a partir de dicho epoch, cuando el aprendizaje se estabiliza en torno al 65% de precisión.

3.3 Arquitectura Resnet152

Una versión del modelo de arquitectura de red residual, donde las salidas de los bloques convolucionales se conectan también directamente con su entradas lo que permite estructuras más profundas sin que se produzca desvanecimiento de gradiente. La versión ResNet 512 se encuentra ya implementada en Keras y es de la que se ha hecho uso para lograr estos resultados:

- Top-1 prediction: 65.66% de acierto.

- Top-3 prediction: 84.50% de acierto.

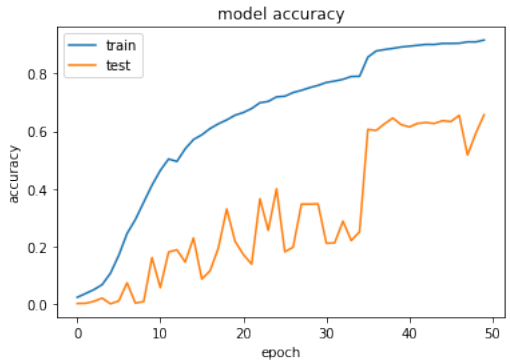

Figure 4: Evolución del aprendizaje de la arquitectura ResNet152

Al igual que sucedía con la arquitectura DenseNet, la evolución del valor de la precisión sobre el conjunto de validación avanza de forma caótica y sin alcanzar ninguna estabilización durante las primeras 30 epochs del experimento, pero a diferencia del experimento anterior, los resultados son ligeramente inferiores.

3.4 Arquitectura adaptada desde otro trabajo similar

En el paper "Insect classification and detection in field crops using modern machine learning techniques" [1] se emplean diferentes técnicas de machine learning aplicadas en relación con la detección y clasificación de insectos en imágenes. Una de las técnicas que se estudian es el empleo de una red convolucional para la clasificación de imágenes en las que aparecen 12 especies distintas de insectos. Dicha red posee la siguiente arquitectura:

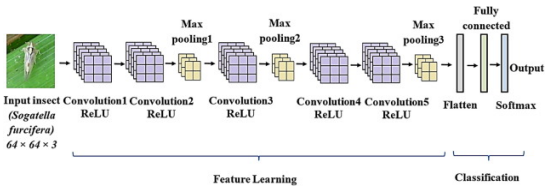

Figure 5: Arquitectura de la red propuesta en el paper

En nuestro experimento se ha replicado manualmente esta red y se ha adaptado su entrada para recibir imágenes de tamaño 32x64, con lo que se han obtenido los siguientes resultados:

- Top-1 prediction: 62.43% de acierto.
- Top-3 prediction: 82.95% de acierto.

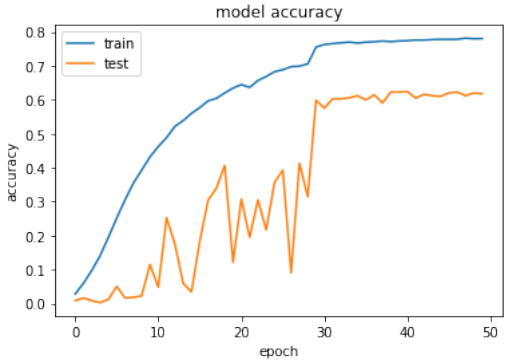

Figure 6: Evolución del aprendizaje de la arquitectura descrita en el paper

En la gráfica puede observarse que, en torno al epoch 30, el aprendizaje comienza a estabilizarse, por lo que los resultados de precisión obtenidos parecen ser consistentes para nuestro dataset. Adicionalmente cabe destacar que pese a ser resultados inferiores a las obtenidos por las redes predefinidas, estas precisiones se consiguen empleando tan solo 370.000 parámetros entrenables en la red, mientras que los modelos DenseNet y ResNet empleados anteriormente requieren 20 y 60 millones de parámetros respectivamente. Esto se traduce en un menor tiempo de entrenamiento y un gasto inferior de recursos de computo.

3.5 Arquitectura propia

En esta sección del trabajo se ha empleado la misma red empleada durante las prácticas de la asignatura para alcanzar un 92% de acierto sobre el dataset CIFAR-100. Dicha red se compone de varios bloques convolucionales, cada uno compuesto por 3 capas convolucionales 2D, cada una a su vez acompañada por una capa de Batch Normalization y una activación ReLu. Estos bloques se repiten 5 veces, comenzando con 32 filtros para las capas convolucionales del primer bloque, y duplicando el número de filtros para cada bloque sucesivo. Por último, al igual que en la red del paper, la salida del último bloque se conecta con una capa Flatten, una capa densa con una activación ReLu y con una capa densa con el mismo número de neuronas que el número de clases de insectos distintas de las que disponemos con una activación Softmax.

Con esta arquitectura se han logrado los siguientes resultados:

- Top-1 prediction: 76.94% de acierto.
- Top-3 prediction: 91.96% de acierto.

La gráfica obtenida presenta una forma considerablemente similar a la obtenida en el experimento anterior, con una estabilización de las precisiones a partir del epoch 30. En este caso, nuestra red obtiene los mejores resultados hasta el momento y de nuevo empleando un número de parámetros entrenables mucho

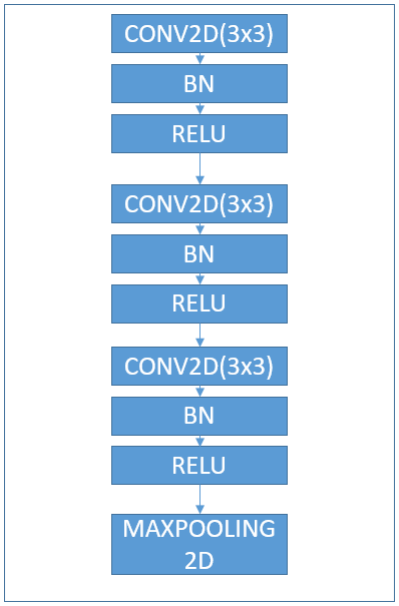

Figure 7: Estructura interna de los bloques convolucionales usados

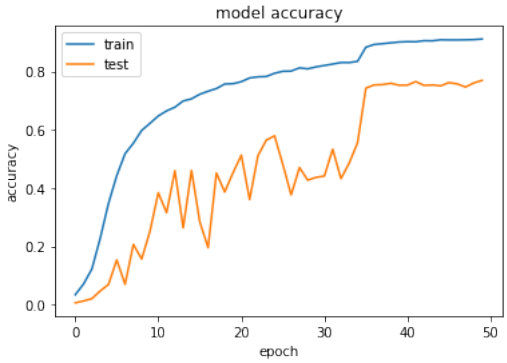

Figure 8: Evolución del aprendizaje de la arquitectura personalizada

más reducido (8.5 millones) que DenseNet, que generaba la mejor precisión hasta el momento.

3.6 Arquitectura propia modificada con ResNet

Empleando técnicas propias de las redes residuales, se ha modificado la arquitectura descrita en el aparatado anterior de forma que la entrada y la salida de los bloques convolucionales se encuentren conectados directamente.

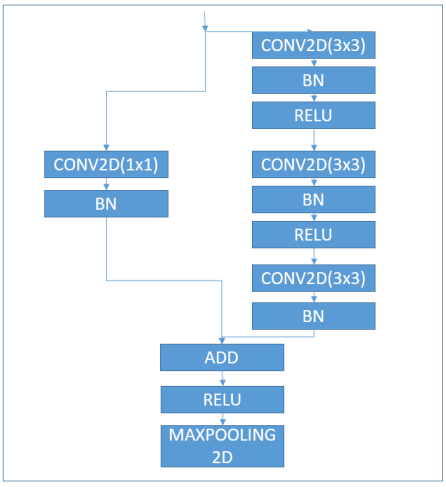

Figure 9: Estructura interna de los bloques convolucionales usados

Con dicha modificación se ha conseguido mejorar ligeramente los resultados:

- Top-1 prediction: 79.75% de acierto.
- Top-3 prediction: 92.63% de acierto.

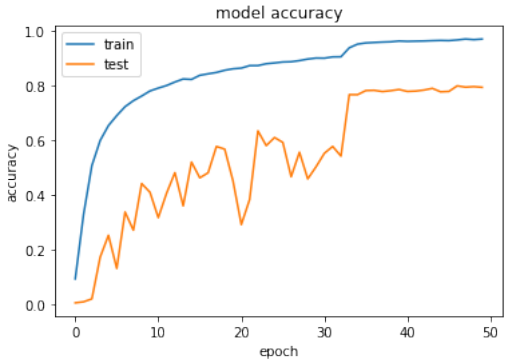

Figure 10: Evolución del aprendizaje de la arquitectura personalizada con ResNet

Siguiendo el mismo patrón que en la red de partida, la modificación con Resnet alcanza una estabilización, tanto para los valores de entrenamiento como para los de validación, en torno al epoch 35.

4 Comparativa

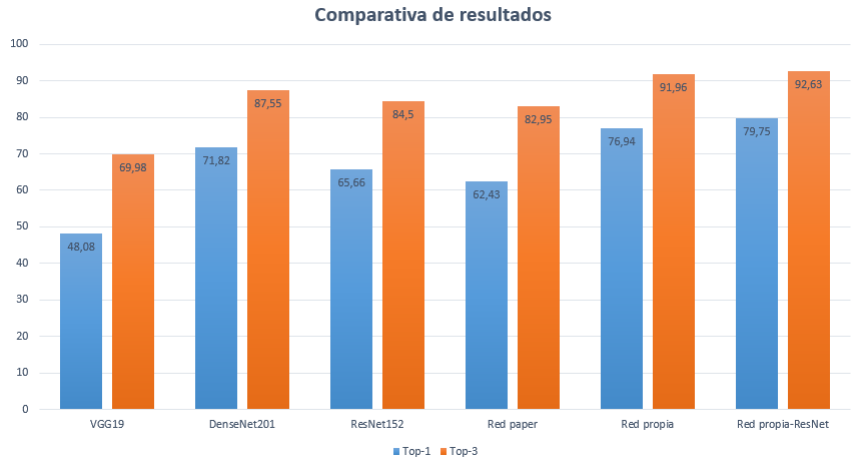

Figure 11: Comparativa de los resultados para cada red

En base a los resultados obtenidos por cada modelo podemos observar como un mayor número de capas y parámetros en un modelo no siempre se traduce en mejores resultados y una mejor adaptación a los datos de entrenamiento. Si bien, como ya se ha comentado, la red adaptada del paper 'Insect classification and detection in field crops using modern machinelearning techniques' no ha obtenido resultados tan satisfactorios como algunas de las redes predefinidas, sí que los ha obtenido notablemente similares a los del modelo ResNet152 con una cantidad menor de capas y profundidad que dicha red.

Lo mismo ha sucedido para nuestra red propia, solo que esta vez los resultados han sido los mejores, con una diferencia del 7% para la variante con técnicas de red residual con respecto al modelo DenseNet201, y con 3 veces menos parámetros entrenables. Esto podría entenderse como una consecuencia de que los modelos predefinidos de Keras han sido generados para resolver problemas de clasificación de ImageNet, donde se deben clasificar 1000 clases de imágenes diferentes, por lo que si bien las 291 clases de insectos con las que se ha trabajado en este problema pueden considerarse elevadas, sigue siendo un número 3 veces inferior a 1000; la misma relación que encontramos entre el número de parámetros de nuestra red y el de DenseNet201.

5 Modelo combinado

Llegados a este punto del trabajo disponemos de distintos modelos que generan distintos resultados sobre el dataset del que disponemos. Por lo tanto, en esta sección se propone combinar los 3 modelos implementados manualmente (el modelo extraído del paper, y las dos variantes de la red propia) en un único

modelo, el cual pasará la misma entrada a cada uno de los 3 modelos y combinará sus salidas en un único resultado.

Para realizar la combinación de las salidas se van a emplear dos tipos de capas de Keraas, Maximun() y Average(). Ya que la última capa de los 3 modelos es una SoftMax() con tantas salidas como clases hay en el dataset, podemos combinar estas salidas calculando, o bien el máximo de cada valor para las 3 redes, o bien calculando la media de los 3 resultados para cada valor, de forma que al final continuemos obteniendo una salida SoftMax con valores entre 0 y 1 y el mismo número de salidas que de clases. Con este planteamiento se han obtenido finalmente los siguientes valores de precisión:

	Top-1	Top-3
Average	80.99%	92.92%
Maximum	81.47%	92.96%

References

[1] Thenmozhi Kasinathan, Dakshayani Singaraju, and Srinivasulu Reddy Uyyala. "Insect classification and detection in field crops using modern machine learning techniques". In: *Information Processing in Agriculture* (2020). ISSN: 2214-3173. DOI: https://doi.org/10.1016/j.inpa.2020.09.006. URL: https://www.sciencedirect.com/science/article/pii/S2214317320302067.