

*ANÁLISIS DE “A PRELIMINARY STUDY ON  
DEEP TRANSFER LEARNING APPLIED TO  
IMAGE CLASSIFICATION FOR SMALL  
DATASETS”*

Por Antonio Jesús García Nieto

## CONTROL DE VERSIONES.

Versión	Descripción	Fecha de última modificación	Validado
1.0	Primera versión estable	25/03/21	No
1.1	Revisión tras feedback. Se han corregido términos y expresiones. Se ha unificado el formato.	31/03/2021	No

## TABLA DE CONTENIDO

Control de versiones.....	2
Introducción .....	4
Precedente .....	4
Metodología .....	5
Fase 1. Adaptación de la red .....	5
Fase 2. Preparación del conjunto de datos .....	5
Fase 3. Planificación de experimentos .....	5
Resultados .....	6
Conclusión .....	7
Bibliografía.....	8

## INTRODUCCIÓN

En este experimento pretendemos comprobar si la red de clasificación binaria propuesta en (Molina et al., 2021) se ajusta a nuestro problema de clasificación múltiple y si es mejor entrenar una red mediante scratching o dividiendo el set de datos y aplicando transfer learning.

## PRECEDENTE

Como hemos comentado el precedente de este experimento es el estudio (Molina et al., 2021).

En él, los autores proponen la siguiente hipótesis:

1. *Un modelo entrenado utilizando técnicas de transferencia del aprendizaje obtiene mejores resultados que el mismo modelo entrenado sobre los mismos datos mediante aprendizaje por raspado.*

Para comprobar su hipótesis inicial, los autores trabajan con un conjunto de datos formado por imágenes de células parasitadas o no parasitadas con malaria, es decir, trabajan en una clasificación binaria, por lo que tendremos que realizar algunas adaptaciones antes de trabajar en nuestro problemas.

Por otro lado, la red propuesta en el proyecto sigue la arquitectura que se muestra en la siguiente imagen:

Layer (type)	Output shape	Params	Updateable
Conv2D	(None, 48, 48, 32)	896	No
Conv2D	(None, 46, 46, 32)	9,248	No
MaxPooling2D	(None, 23, 23, 32)	0	No
Conv2D	(None, 21, 21, 64)	18,496	No
MaxPooling2D	(None, 10, 10, 64)	0	No
Flatten	(None, 6400)	0	No
Dense	(None, 128)	819,328	Yes
Dense	(None, 2)	258	Yes

Dos detalles importantes de la implementación son, el uso de kernel de tamaño 3x3 para todas la capas de convolución y que el tamaño de filtro de las capas de max pooling es de 2x2.

Antes de trabajar con esta arquitectura vamos a destacar su parecido con la familia de redes VGG, propuestas en el estudio (Simonyan & Zisserman, 2015). Ciertamente parece una versión reducida de esta familia de redes.

La red SOCO explota una idea vista por primera vez en las redes de la familia VGG, el uso kernels de convolución uniformes de tamaño con filtro 3x3 apilados. Como se detalló en la sección de arquitecturas cuando se habló sobre la familia VGG, el usar este tamaño de kernels y apilar capas de convolución genera el mismo efecto que aplicar kernels más grandes y además añade algunas ventajas como ser computacionalmente más ligeras (siendo C el número de canales, para 3 capas de convolución con un filtro 3x3 tendríamos  $3^2 \cdot C^2$  o lo que es lo mismo  $27C^2$  parámetros, mientras que para una capa convolucional con un kernel de 7x7 tendríamos  $49 \cdot C^2$  parámetros). Además, el apilar varias genera no linealidades extras (mientras que con un kernel de 7x7 se genera una, con 3 capas con kernel de 3x3, que es el equivalente, generamos 3).

Una idea que no han tomado de la red VGG y que quizás pudiesen ser beneficiosas es añadir una convolución con un kernel de tamaño 1x1, esto sirve para general no-linealidades extra sin afectar al tamaño del mapa de características y además permite realizar una condensación de la información que realmente no empeora la precisión de la red como se describe en (He et al., 2016).

## METODOLOGÍA

En este estudio queremos tratar dos hipótesis:

1. *¿Podrá la red SOCO servir para nuestro problema de clasificación multiclase?*
2. *¿Será beneficioso el usar transfer learning en esta red para problemas multiclase?*

Para saber si se cumplen las premisas anteriores trabajaremos en tres fases:

### FASE 1. ADAPTACIÓN DE LA RED

En primer lugar replicamos la red (el código de la implementación se encuentra en el repositorio de GitHub) y la adaptamos haciendo unas ligeras modificaciones como sustituir la capa de salida (recordemos que en (Molina et al., 2021) el problema que se aborda es una clasificación binaria, mientras que nosotros vamos a trabajar en un problema multiclase).

En el estudio, la red propuesta tiene una capa de salida con solo una neurona, con una función de activación sigmoide, lo cual es apropiado para una clasificación binaria, nosotros usaremos una capa de 7 neuronas (una por clase) y una función de activación softmax.

La red recibirá el nombre de CNN\_SOCO y la entrenamos con los mismos hiperparámetros que en (Molina et al., 2021):

- Optimizador RMSProp con un ratio de aprendizaje de 0.0001.
- Tamaño de lote de 128.
- 5 épocas.

### FASE 2. PREPARACIÓN DEL CONJUNTO DE DATOS

Para esto, dividiremos el set de datos en subconjuntos y compararemos el resultado con el obtenido en el punto anterior.

Para ello separaremos los datos en dos sets:

- Source: Consiste en la información que se usará para entrenar completamente el modelo. Consiste en un 60% del total del set de datos.
- Target: Consiste en la información que se usará para entrenar las capas de salida del modelo. Se divide a su vez en train (70%) y test (30%)

### FASE 3. PLANIFICACIÓN DE EXPERIMENTOS

Para comparar los resultados que obtengamos con los del paper vamos a hacer los mismos 4 experimentos que ellos llevan a cabo.

- Caso 1 - CE = train, CT = test
- Caso 2 - CE = source, CT = test
- Caso 3 - CE = source + train, CT = test
- Caso 4 - CE = source, CRE = train, CT = test

Donde CE es conjunto de entrenamiento, CRE conjunto de reentrenamiento, CT conjunto de test.

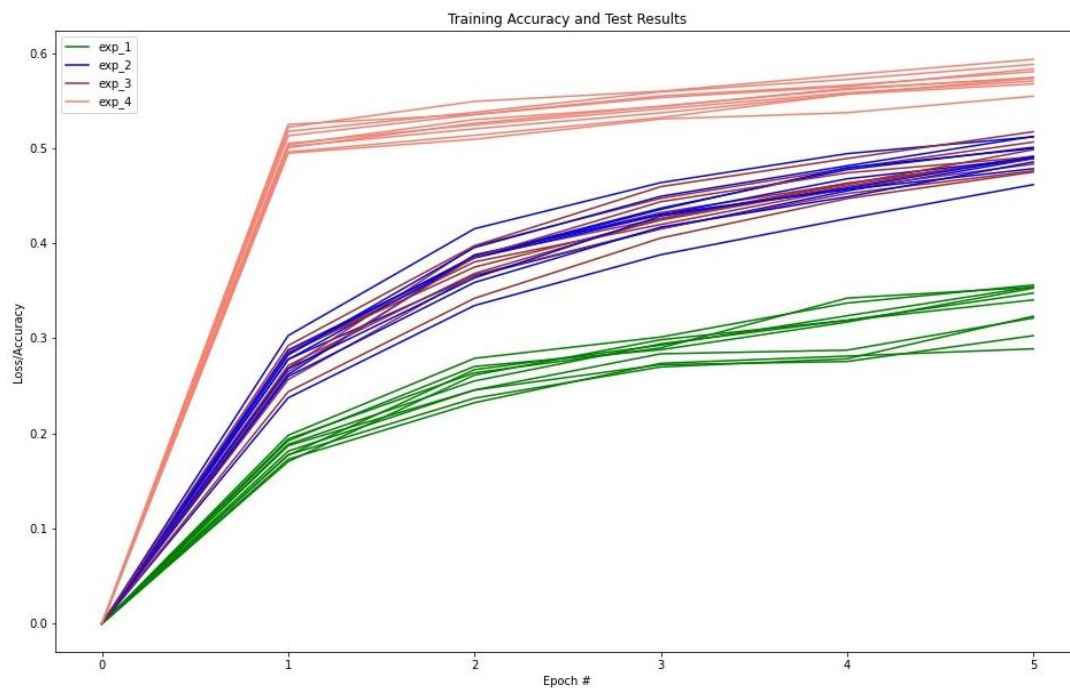
Para demostrar que los resultados son concluyentes se realizarán 10 iteraciones con cada configuración.

## RESULTADOS

En la siguiente tabla podemos ver los resultados obtenidos tras testear el modelo.

Iteración	Caso 1	Caso 2	Caso 3	Caso 4
1	0.371633	0.348367	0.402857	<b>0.432449</b>
2	0.365510	0.381837	0.362653	<b><u>0.446531</u></b>
3	0.394082	0.464898	0.376939	<b><u>0.473878</u></b>
4	0.334694	0.313061	<b><u>0.370000</u></b>	0.337959
5	0.336939	<b><u>0.492653</u></b>	0.401224	0.428163
6	0.325714	0.313469	0.358775	<b><u>0.443673</u></b>
7	0.304082	0.449184	0.283061	<b><u>0.503878</u></b>
8	0.319388	<b><u>0.398367</u></b>	0.260816	0.350612
9	0.255918	0.327755	0.431633	<b><u>0.454898</u></b>
10	0.330408	0.367755	0.264082	<b><u>0.481020</u></b>
Media	0.332551	0.374796	0.366327	<b><u>0.435306</u></b>

También podemos ver los resultados sobre el set de datos de entrenamiento en forma de gráfico.



## CONCLUSIÓN

Podemos obtener varias conclusiones, si bien es cierto que los resultados son notablemente mejores cuando usamos técnicas de transferencia del conocimiento, también podemos darnos cuenta de que los resultados no son demasiado buenos.

La mejora media con los métodos de transfer learning sobre el siguiente mejor resultado es de aproximadamente el 16%. Luego tendremos que recordar esto de cara a un futuro trabajo.

Por otro lado, la precisión en el mejor de los casos es del 0.503878 %. Dado que nuestro objetivo consiste en el diagnóstico de enfermedades, esta precisión no es aceptable. No obstante, basado en otros estudios, podrían añadirse algunas mejoras tanto a nivel arquitectónico (hemos citado alguna en la sección Precedente) como a nivel de hiperparámetros.

Una forma de mejorar la red solo modificando los hiperparámetros, sería modificar el tamaño del lote. En algunos estudio como (Kandel & Castelli, 2020) se pone de relieve que no es buena idea usar tamaños de lote altos y ratios de aprendizaje bajos. Sería interesante probar lotes con un tamaño de 32.

En definitiva, podemos confirmar que se cumple la primera hipótesis, pero no podemos comprobar que se cumpla la segunda.

## BIBLIOGRAFÍA

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6(4), 312–315. <https://doi.org/10.1016/j.ict.2020.04.010>
- Molina, M. Á., Asencio-Cortés, G., Riquelme, J. C., & Martínez-Álvarez, F. (2021). A Preliminary Study on Deep Transfer Learning Applied to Image Classification for Small Datasets. *Advances in Intelligent Systems and Computing, 1268 AISC*, 741–750. [https://doi.org/10.1007/978-3-030-57802-2\\_71](https://doi.org/10.1007/978-3-030-57802-2_71)
- Simonyan, K., & Zisserman, A. (2015, September 4). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. <http://www.robots.ox.ac.uk/>