

APLICACIÓN DE ARQUITECTURAS CONVENCIONALES A LA CLASIFICACIÓN DE LESIONES DE PIEL

Por Antonio Jesús García Nieto

CONTROL DE VERSIONES

[illegible]

TABLA DE CONTENIDO

Control de versiones.....	2
Introducción	4
Planteamiento del estudio	4
Resultados	¡Error! Marcador no definido.
Conclusión del experimento.....	¡Error! Marcador no definido.
Bibliografía.....	¡Error! Marcador no definido.

INTRODUCCIÓN

En este experimento se trabajarán varios conceptos interesantes dentro del mundo de las redes neuronales, como son el transfer learning (ya tratado en el primer experimento), el fine tuning (o ajuste fino) y el uso de arquitecturas clásicas de CNN.

Antes de enunciar nuestra premisa definiremos brevemente los conceptos que son la base de este estudio:

- **Transfer Learning.** Consiste en usar modelos ya entrenados con grandes cantidades de datos. Esto nos permite no dedicar recursos ni tiempo en entrenar estos modelos. Lo más común es congelar la mayor parte de la red, dejando que solo las capas finales y clasificadores se entrenen.
- **Fine Tuning.** Es una práctica que se suele aplicar después del proceso de transfer learning. Consiste en descongelar el modelo, entrenarlo de forma completa con un ratio de aprendizaje mucho más bajo. Esta técnica permite ajustar la red ligeramente, pero sin romper los patrones aprendidos con anterioridad.
- **Arquitecturas convencionales.** Ya se han tratado algunas en la sección “”. Estas son básicamente arquitecturas que han sido importantes o que actualmente son muy usadas.

Las hipótesis que pretendemos validar en este estudio son:

1. *¿Las arquitecturas clásicas como VGG-16, VGG-19, ResNet-50, EfficientNet ... darán una precisión mayor que nuestra red ...?*
2. *¿Las arquitecturas clásicas guardan una relación aceptable entre precisión y rendimiento?*

PLANTEAMIENTO DEL ESTUDIO

El estudio consistirá en una comparación de la precisión media entre distintas redes como: VGG-16, ResNet50, Inception y RSM_BAJ.

La configuración general será:

Se seguirá el mismo esquema de entrenamiento que en el experimento 1. Recordemos que el este es:

- Los datos serán 10k imágenes de un tamaño de 128x128.
- El conjunto de entrenamiento = source (60% del total del conjunto de datos),
- El conjunto de reentrenamiento = train (70% del 40% restante del conjunto de datos),
- El conjunto de test = test (21% del 40% restante del conjunto de datos)
- El conjunto de validación = validación (9% del 40% restante del conjunto de datos)

Para las redes preentrenadas se entrenará el modelo con source y train. Para la red RSM_BAJ, se utilizará la técnica de preentrenamiento usada en otros estudios y propuesta en (Molina et al., 2021)

Los hiperparámetros serán los siguiente:

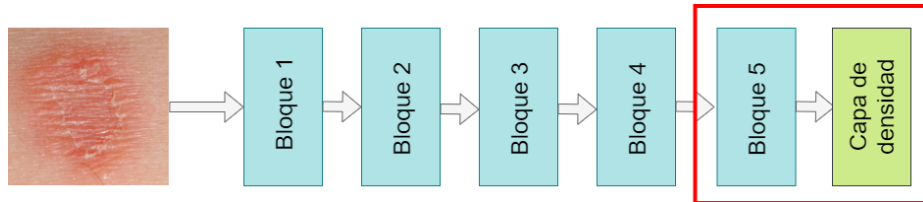
- Ratio de aprendizaje de 0.0001, durante el Transfer Learning.
- Ratio de aprendizaje de 0.00001, durante el Fine Tuning.
- Épocas: 20 para la fase de transfer learning y para la de fine tuning.
- Callbacks: Se detiene el entrenamiento si la red consigue peores resultados en 3 ocasiones consecutivas.
- Tamaño de lote: 16
- Optimizador Amsgrad

ARQUITECTURAS VGG

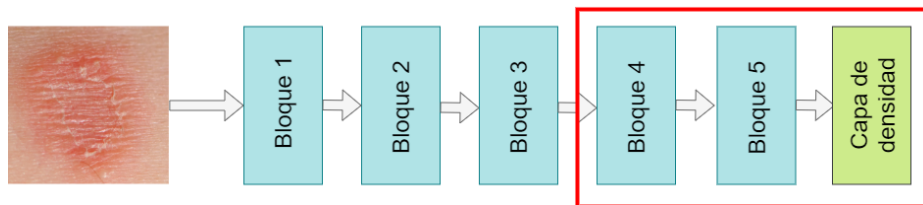
Como hemos visto en la sección de arquitecturas históricas las redes VGG están compuesta por 5 bloques convolucionales. Para trabajar con estas arquitecturas trabajaremos con los pesos de ImageNet.

Realizaremos dos experimentos:

Primer experimento: congelando los bloques 1, 2, 3 y 4 e introduciendo una capa de densidad nueva. Posteriormente aplicamos el fine tuning.



Segundo experimento: congelando los bloques 1, 2 y 3 introduciendo una capa de densidad nueva. Posteriormente aplicamos el fine tuning.



ARQUITECTURA RESNET-50

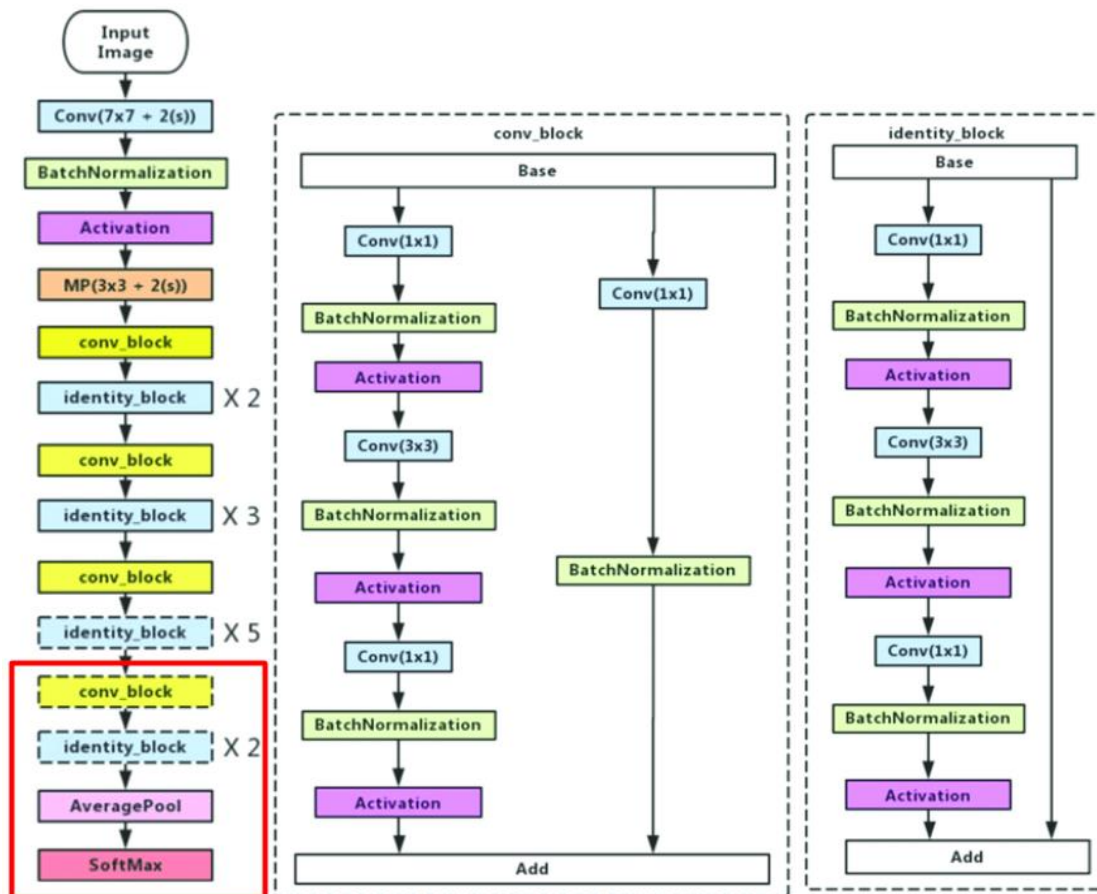
ResNet50, como vimos en la sección de análisis, es una arquitectura con una forma característica, para este estudio trabajaremos con su ultimo bloque convolucional. Denominamos bloque convolucional al conjunto de los dos últimos bloques de identidad, como podremos observar en la imagen de la próxima página está formado por 2 bloques de identidad y un bloque convolucional, ambos bloques también vienen definidos. y un bloque convolucional.

Los experimentos que realizaremos serán los siguientes:

Primer experimento: congelaremos el ultimo bloque de identidad y el clasificador. Después aplicaremos fine tuning.

Segundo experimento: congelaremos los dos bloques de identidad y el clasificador. Después aplicaremos fine tuning.

Tercer experimento: congelaremos el bloque de convolución, los dos bloques de identidad y el clasificador. Después aplicaremos fine tuning.



ARQUITECTURA INCEPTION

Respecto a Inception, realizaremos dos experimentos.

Primer experimento: congelaremos el ultimo bloque de identidad y el clasificador. Después aplicaremos fine tuning.

Segundo experimento: congelaremos los dos bloques de identidad y el clasificador. Después aplicaremos fine tuning.

Tercer experimento: congelaremos el bloque de convolución, los dos bloques de identidad y el clasificador. Después aplicaremos fine tuning.

ARQUITECTURA RSM_BAJ

RSM_BAJ es la arquitectura propuesta en el experimento 3. En este caso, debido a que no es una arquitectura preentrenada trabajaremos con ella sin realizar transfer learning.

RESULTADOS

Arquitectura	ResNet50			Inception		VGG16		RSM_BAJ
Bloques entrenados en la fase de TF	1	2	3	0	1	1	2	Todos
Transfer Learning	68.1%	26.36%	17.59%	92.7%	91.08%	95.64%	17.46%	95,39%
Fine Tuning	95.0%	95.82%	94.23%	93.46%	92.48%	95.89%	95.82%	95,33%

En la tabla anterior se resumen los resultados medios de cinco entrenamientos de cada modelo obtenidos para cada experimento.

CONCLUSIONES

Como conclusión en este estudio podemos confirmar que el fine tuning es prácticamente obligatorio para optimizar el rendimiento de los modelos preentrenados. En caso de querer usar solo transfer learning la opción más robusta parece ser VGG. Sin embargo, cabe destacar la diferencia de parámetros entre las 4 redes VGG16 cuenta con 15,7M de parámetros, ResNet50 27,7 M, InceptionV3 23,8M y RSM_BAJ con tan solo 8,6 millones.

Teniendo en cuenta lo expuesto anteriormente y la diferencia entre la precisión de los modelos parece ser que RSM_BAJ es una opción a tener en cuenta la cual reduce significativamente el número de parámetros.

