UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

Departamento de sistemas informáticos y computación

Computer Vision

Ejercicios propuestos

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Autor
Francisco Javier Gil-Terrón Rodríguez

2021 - 2022

Tabla de contenidos

1. Gender recognition

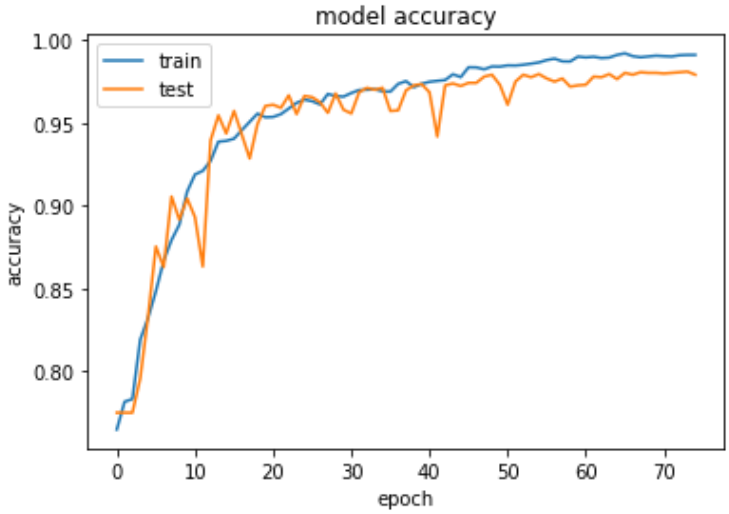
Esta primera tarea consiste en la clasificación de imágenes de caras en función del género (hombre o mujer), por lo que se trata de una tarea de clasificación en dos clases.

Para llevarla a cabo se han utilizado 10.585 imágenes para entrenamiento y 2648 para test, las cuáles han sido normalizadas al comienzo de la tarea.

Hay dos objetivos principales para este caso: alcanzar 97% de accuracy y alcanzar 92% pero con una red que tenga menos de 100.000 parámetros.

Para ambos casos se ha trabajado sobre TensorFlow y Keras y se ha utilizado SGD como algoritmo optimizador con factor de aprendizaje inicial 0.1 y ReduceOnPlateau para reducirlo, y entropía cruzada como función de pérdida. Se ha utilizado un batch size de 100 durante 75 epochs y como data augmentation se ha trasladado en la vertical y horizontal un 10%, se ha usado flip horizontal, zoom de un 20% y se han usado 3° de rotación.

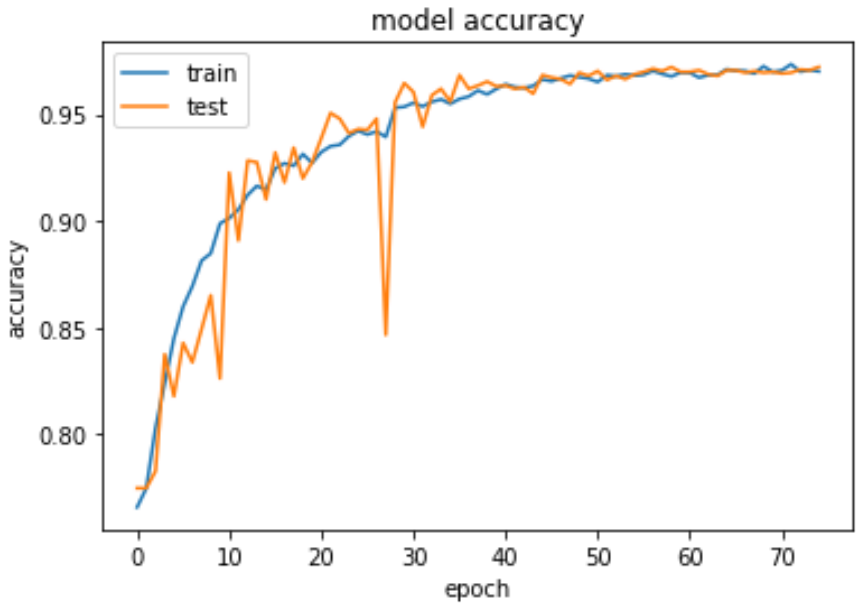
Para el primer caso, se ha utilizado una red compuesta por cinco bloques de convolución (kernel 3) + Batch normalization + ruido gaussiano + ReLu como función de activación + MaxPool (size 2) con 32, 64, 128, 256 y 512 filtros respectivamente. Tras esto, se conecta con una capa densa de 512 + ReLu y finalmente la densa con el número de clases (2) + softmax. El total de parámetros de esta red es de 3.933.378.



Con esta configuración el mejor resultado que se ha obtenido sobre test llegaba a un 98.11% de accuracy, cuya gráfica de resultados se refleja la anterior imagen.

Para el segundo caso, se han utilizado bloques convoluciones y deconvolucionales (kernel 3) con Batch normalization, ruido gaussiano y ReLu como función de activación. Así pues, la red comienza con 4 bloques convolucionales con 10, 20, 40 y 80 filtros, y tras esto se emplean otros 4 bloques deconvolucionales con 40,20,10 y 3 filtros. Cabe destacar que en estos bloques no se hacía MaxPool, por lo que tras estos bloques se utiliza un AveragePool de tamaño 2 para finalmente conectar a una densa con el número de clases + softmax. Con todo esto se obtiene una red de 92.257 parámetros, que cumple con el requisito que se proponía.

A continuación, se muestra la gráfica de resultados obtenidos por esta red (el resto de las configuraciones es idéntica a la empleada en la anterior), donde se ha alcanzado una accuracy máxima de 97.21%, con lo que incluso con esta red más ligera se lograba el punto anterior.



2. Colorization

En este segundo ejercicio se plantea crear un modelo capaz de devolver imágenes a color a partir de una entrada en blanco y negro. Una vez más, se ha utilizado TensorFlow y Keras, y se ha trabajado sobre un dataset de 40.000 imágenes variadas (sin una categoría concreta), de las cuales se han extraído 1.000 aleatorias para entrenamiento, y 100 para test, y se han normalizado en un inicio.

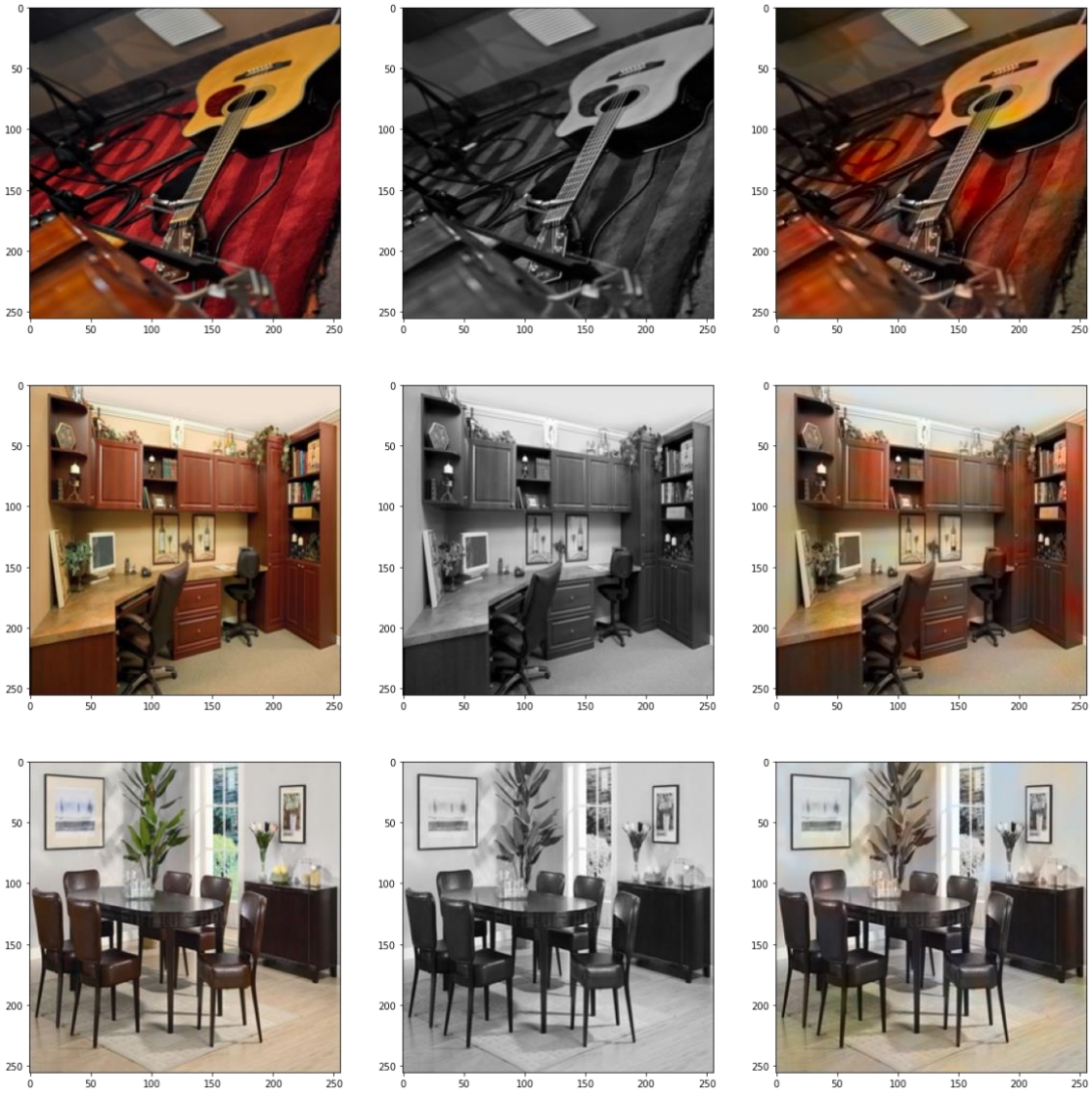
La red que se ha utilizado era la propuesta en el GitHub de la asignatura: un modelo con doble entrada para las imágenes y vector obtenido con un modelo Inception pre-entrenado. La red sigue una estructura de encoder-decoder con bloques upsampling en el decoder

La configuración de data augmentation ha sido zoom 40%, shear 40%, rotación 40º y flip horizontal, aunque debido a la naturaleza de la tarea no es demasiado relevante esta configuración. El algortimo optimizador empleado ha sido Adam con factor inicial de 0.001 y ReduceOnPlateau con paciencia 10 pero factor multiplicativo de 0.5 para suavizar la convergencia, puesto que el fin de que los colores sean más vivos la red se ha dejado entrenar durante 500 epochs (ya que la función de activación de salida es una tangente hiperbólica).

La función de pérdida utilizada es el error cuadrático medio, pero no se mostrarán los resultados de la loss obtenidos gráficamente ya que, además de ser dependiente del conjunto de datos utilizado, podría no reflejar correctamente los resultados obtenidos, ya que la correcta (o no) colorización de una imagen dependería más del usuario que de un criterio objetivo. No obstante, sí que se comentará que durante las 500 epochs esta loss se redujo de 0.01 hasta 0.0059.

En su lugar, a continuación, se mostrarán algunas de las imágenes que han sido coloreadas por el sistema. La primera corresponde a la original, la segunda el input de la red, y la tercera el output.

Como se puede ver, no logra alcanzar la nitidez de los colores más vivos, aunque gracias a dejarlo durante 500 epochs, se ha logrado salir de la tonalidad sepia y neutra con la que colorea la red en las primeras epochs.



3. Segmentation

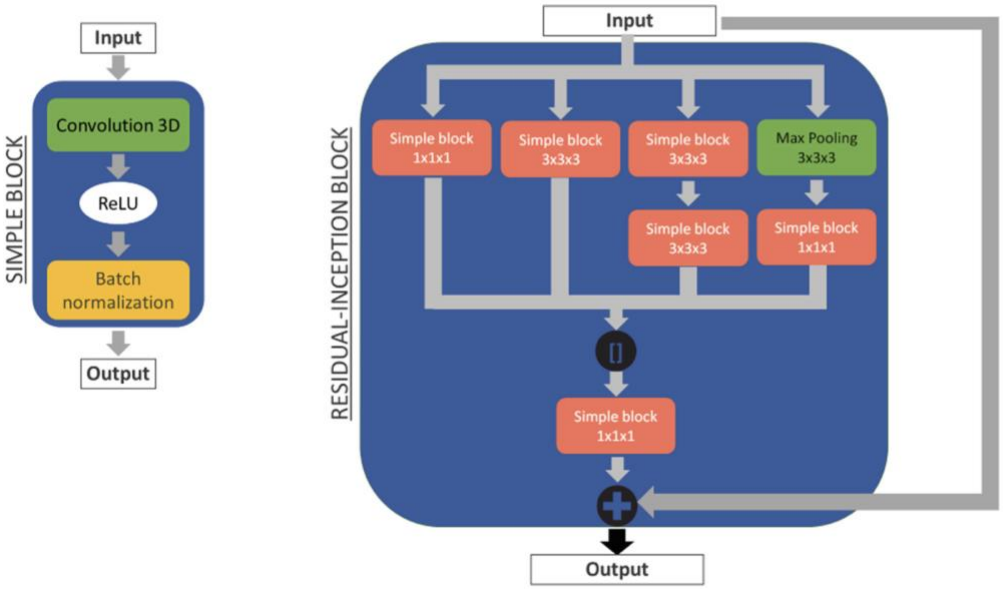
Esta última tarea tiene como objetivo la segmentación de vasos sanguíneos del ojo humano. Al contrario que en los casos anteriores, se ha utilizado PyTorch y Monai, framework de Deep learning enfocado en imagen médica que, aunque no sea exactamente el propósito para el que está pensado (ya que utiliza resonancias magnéticas mayoritariamente) será de gran ayuda para el data augmentation.

El dataset se compone de 20 imágenes de ojos humanos donde se diferencian los vasos sanguíneos. Se han utilizado 19 para el entrenamiento, y 1 para test, que, de nuevo, han sido normalizadas al comienzo de la tarea.

La función de pérdida que se ha utilizado es Dice y el algoritmo optimizador Adam con learning rate inicial de 0.001 y scheduler ReduceOnPlateau.

En cuanto al data augmentation, se han utilizado variaciones de intensidad del 10% y desplazamientos del 10% en aquellas zonas de mayor intensidad (es decir, los vasos sanguíneos).

La red que se ha utilizado es una Residual Inception U-net. Este tipo de red se basa en bloques Residual Inception con el objetivo de capturar características a diferentes escalas. Estos bloques son:

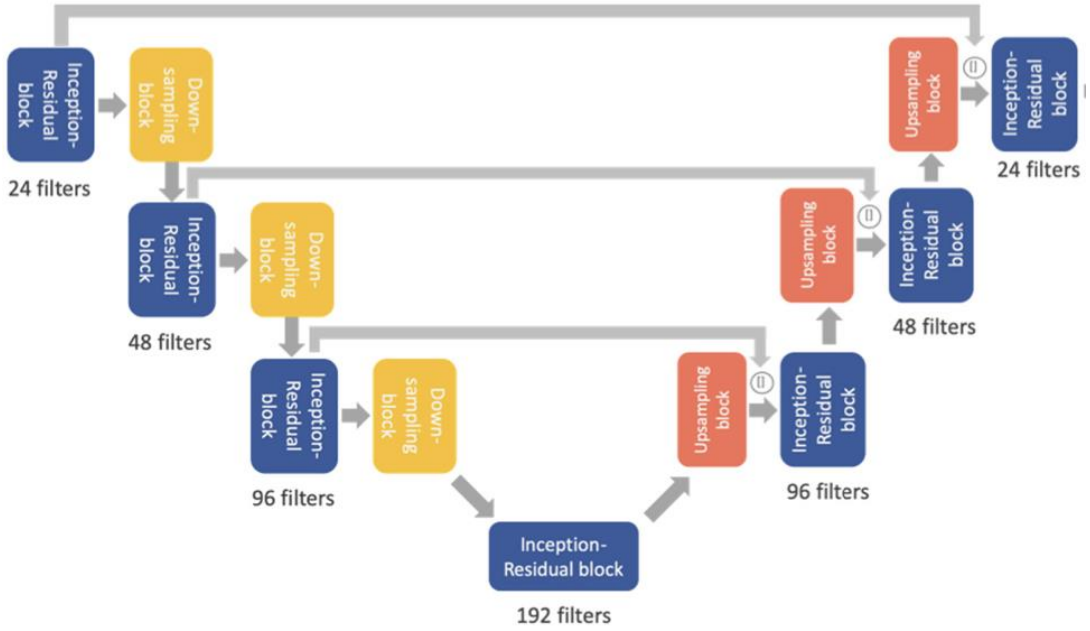


Para simplificar, el término simple block se utiliza para denotar que se está utilizando una convolución seguida de una función de activación ReLu, y batch normalization.

Estos bloques Residual Inception estarán compuestos, como se ilustra en la figura anterior, con la estructura:

- Bloque simple con tamaño de kernel 1.
- Bloque simple con tamaño de kernel 3.
- Dos bloques simples consecutivos con tamaño de kernel 3.
- Max Pooling con tamaño de kernel 3 (y stride 1) seguido de un bloque simple con tamaño de kernel 1.

La salida de estos cuatro canales se concatena y después se aplica otro bloque simple con tamaño de kernel 1 para reducir la información. Por último, una conexión residual une la entrada del bloque Residual Inception con la salida. El número de filtros utilizado en cada uno de estos bloques dependerá del nivel de la red en el que se encuentre, como se describe a continuación.



Como se puede observar en la figura anterior, el diseño final consta de cuatro niveles de profundidad, donde, en el caso del encoder (downsampling), se utilizan bloques convolucionales de tamaño de kernel 3 con stride 2, ReLu como función de activación y Batch Normalization, y en el caso del decoder (upsampling) bloques deconvolucionales o de convolución transpuesta con las mismas características.

Cada uno de estos niveles tiene, respectivamente, 24, 48, 96 y 192 filtros, donde se aplica un bloque Residual Inception con el número de filtros de su nivel correspondiente.

Además, siguiendo la estructura de la red en U, se utilizan conexiones de entre niveles simétricos para mejorar el flujo de gradiente a través de la fase de entrenamiento.

El modelo se ha entrenado durante 1000 epochs con lo que se ha alcanzado finalmente un 93% de Dice. Cabe destacar que las primeras 200 epochs, pese a que la loss mejoraba, el Dice no lo hacía, debido a que clasificaba la imagen entera con la misma clase hasta que el modelo aprendió lo suficiente como para especializarse en los vasos sanguíneos, por lo que a partir de las 200 epochs el Dice comienza a mejorar.

A continuación, se muestran la imagen de test con su etiqueta, y finalmente el output de la red. Como se puede ver la segmentación se aproxima considerablemente al ground truth, aunque se pueden ver diferencias de grosor en las raíces de los vasos sanguíneos.