He tratado de mejorar la precisión del modelo a través de lo que se le conoce multi-input model, esto significa la entrada al modelo es la imagen y datos extras sobre la imagen, en este caso información de los colores de la imagen.

Entrenamiento

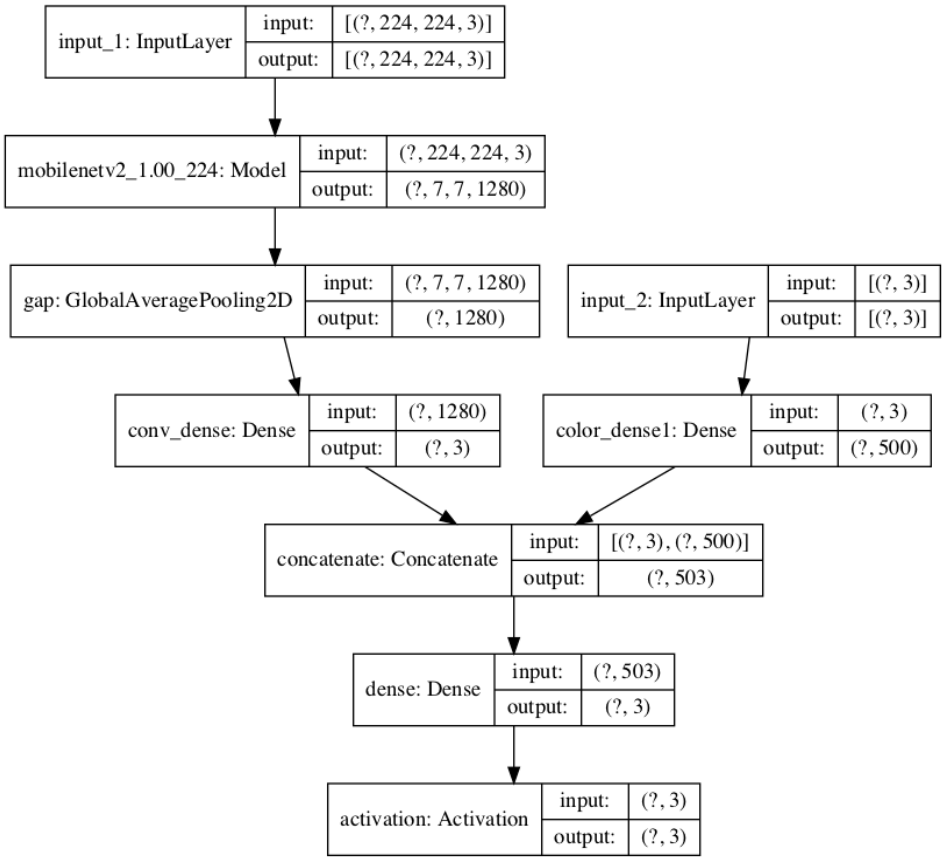
La forma en la que se entrena el modelo es un poco diferente a modelos anteriores, el problema era que el entrenamiento era muy inestable y la precisión en el dataset de entrenamiento llegaba muy rápido a 1.0.
Para las siguientes pruebas también utilizó transfer learning pero solo entreno la ultimo capa por los primeros 20 epochs, y otros 20 epochs entrenando de la capa 75 en adelante. Esto permite obtener resultados más estables y es más fácil ver si realmente mejoró el modelo, con la desventaja que el entrenamiento es más tardado.

El modelo base con el que se hacen las comparaciones tuvo una precisión en el dataset de prueba de 0.79,

Color RGB

En este caso el modelo toma como entrada la imagen y un arreglo de tres valores que representan el color de la fruta en RGB. Por ejemplo, una imagen de un plátano tendría el valores de [1.0, 1.0, 0.0] y una naranja [1.0, 0.64, 0.0].

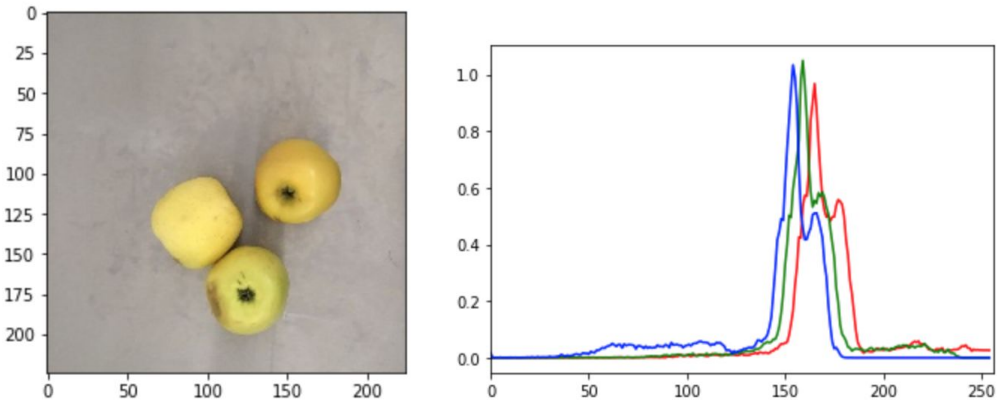
El mejor modelo entrenado tuvo una precisión de 0.90, mejoro por una diferencia de +0.11. La arquitectura del modelo es la siguiente:



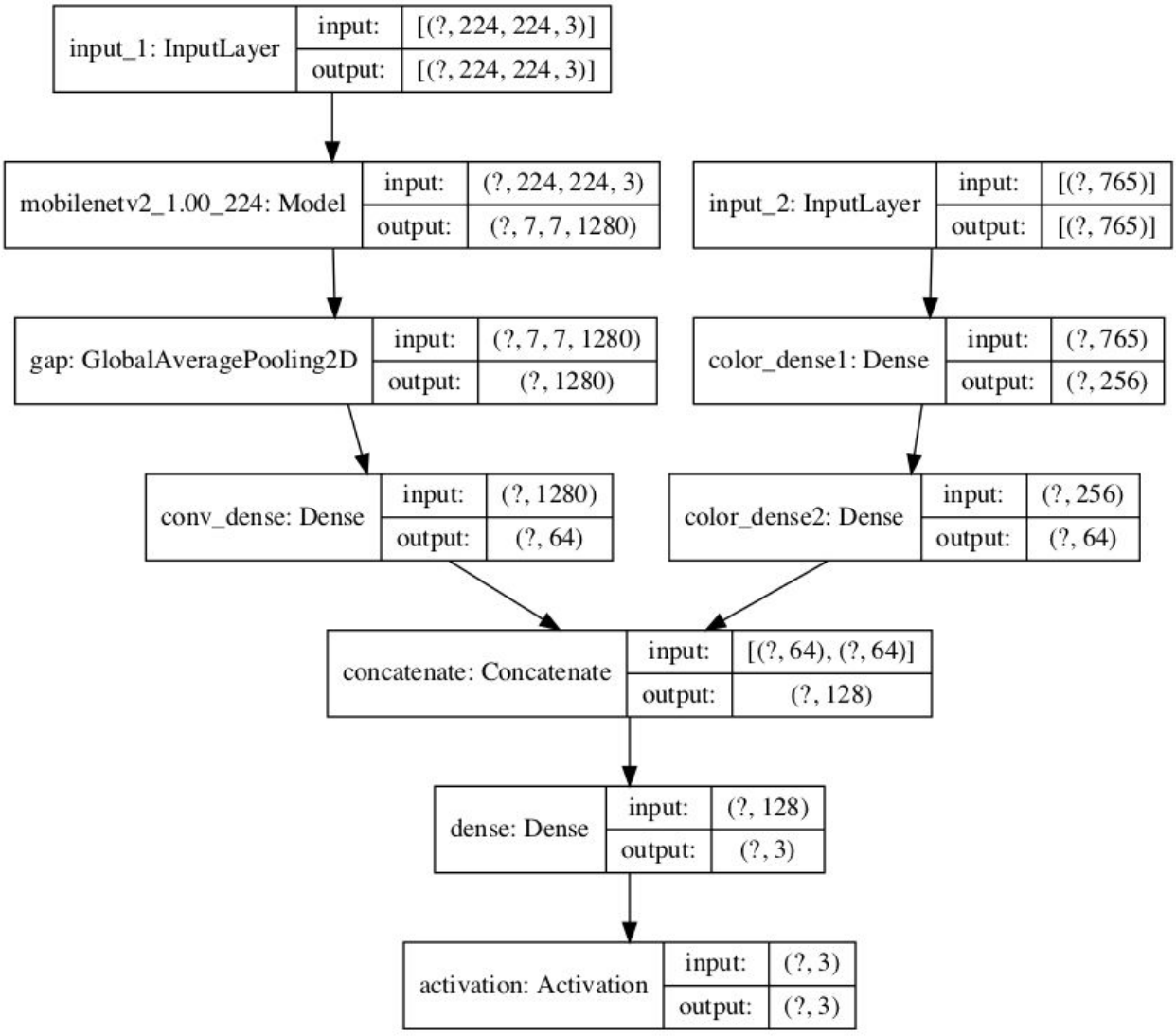
El unico detalle es que no tenemos manera de obtener el valor del color de forma precisa, en este caso lo hice a mano porque ya se por adelantado que color es, pero prueba que si puede mejorar la precisión si tenemos más información sobre la imagen.

Histograma

Otra forma de obtener información acerca de la imagen es a través del histograma, por ejemplo:

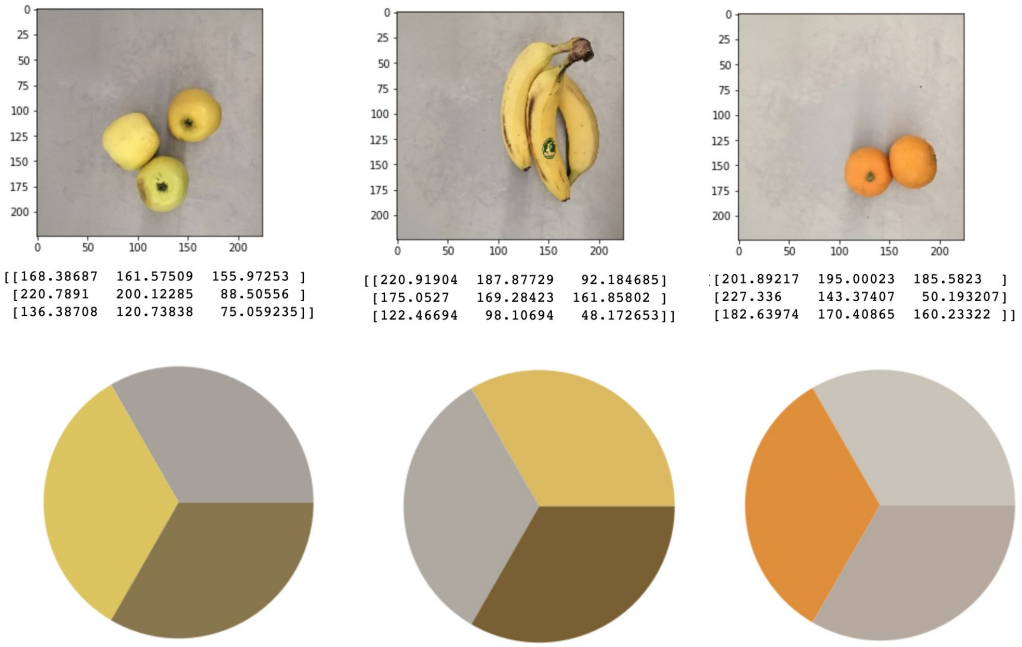


De esta manera los valores del histograma que son 765 son la otra entrada al modelo. Lo malo de usar el histograma es que la mayoría de los valores son 0, lo que causa que propaguen valores muy bajos a través de la red neuronal. Usando el histograma obtuve una precisión de 0.84 mejoro por una diferencia de +0.5. La arquitectura del modelo es la siguiente:



K-Means

Otra forma de obtener información acerca de los colores de una imagen es usando el algoritmo de clustering K-Means. Y se puede implementar en Tensorflow lo cual nos permite, utilizarlo como si fuera una capa de la red neuronal. Con el algoritmo obtenemos los siguientes colores:



Usando el algoritmo de K-Means obtuve una precisión de 0.86 mejoro por una diferencia de +0.7. La arquitectura del modelo es la siguiente: