Structure of Image classification

Dataset url

Set constants etc = epochs, batch size, image dimensions

Split data set, set what train is, parameters on preprocessing.image\_dataset\_from\_directory

Split for validation- K fold validation, split into chunks, and ratio wise test the model on the chunks recursively.

Standardise data, since original RGB are 0,255. Standardise to 0,1 to make input small. Can be done using rescaling, then apply either using map, or include layer in model definition

Create model- layer with 128 unit on top activated by relu

Train model

With a less training examples, overfitting is a problem, solved with data augmentation and dropouts. Augmentation when small number of training examples. Done by taking existing data then using random transformation to yield similar images. Implemented with preprocessing. Can also do custom DA layers. Started with lambda layer. Then, subclass a new layer, both will invert colours to probability. Can also use tf.image and tf.data for fine tuning.

**Data augmentation** can also involve flipping v/h as well as greyscaling/saturating, changing brightness, cropping, rotating. Applied using Dataset.map

Dropout is a regularisation technique. Randomly drops out with activation set to 0, output units. Takes fractional number as input value.

Predict new data

In data augmentation, use preprocessing layers to resize images to consistent shape, and rescale pixel values. Can use by making layers part of model-where it will run synchronously, benefits from GPU accel, as well as when exporting model with model.save, the layers are saved. This means it will automatically standardise images.

Can also apply layers to dataset where it will happen asynchronously. Can overlap training with data preprocessing.

What is image classification

Supervised learning.

Define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos. Early computer vision models relied on raw pixel data as the input to the model. However, as shown in Figure 2, raw pixel data alone doesn't provide a sufficiently stable representation to encompass the myriad variations of an object as captured in an image. The position of the object, background behind the object, ambient lighting, camera angle, and camera focus all can produce fluctuation in raw pixel data; these differences are significant enough that they cannot be corrected for by taking weighted averages of pixel RGB values.