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

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

Predict new data