



Intro to Computer Vision

1. Grey scale pictures reduces the image complexity. So in case, where the color of the image does not matter, example determining is it's a shoe or a bag, you can use a gray scale image instead of an RGB image.
2. **Label** - describes the image, since computers do better with numbers than text. Also, helps us reduce the bias, for example, if we label it as shoes, we are being biased towards the English language. Instead, we can use a label and convert the output into required format/language.
3. While building a NN for a dataset:
 - a. Number of classes/prediction categories = no. of output layer neurons
 - b. Input shape = size of the image, ex: (28x28), then `input_shape = (28, 28)`
 - c. Flatten (Function of Layer class of Keras) : Takes the input shape and converts into a linear array.
4. **Callbacks**: Its a special function, where you can program when to stop the model training. For example, if `loss < 0.4: model.stop_training = True`. This function can then be instantiated and used as a parameter for callbacks in the `model.fit()` function. Now usually, the callback class should have **`on_epoch_end()`** function. This should mean, after every epoch, when you get the log information, check a particular value, in this case, loss value. That's because, during an epoch the

values can vary considerably and hence it is safer to check the logs after every epoch and not during the epoch iteration.

5. **ReLU Activation Function:** Returns x only if $x > 0$ or returns 0.
6. **Softmax Activation Function:** It takes a list of values and scales them such that all the values sum up to 1. After applying this function in the output layer, all these scaled values can be thought as the probabilities for each output class. And the class/category having the highest probability will be the result of the classification.