Organisation

- Train, Val and Test data in separate folders, and each class is in its own folder

ImageDataGenerator

- This is the method used by **tensorflow**
- **Method 1 -** Using dataset_from_directory
  - `ds_train = tf.keras.preprocessing.image_dataset_from_directory(… … …)`
  - Augmentations with `tf.image.augmentation_name(x, ...)`
  - Apply augmentation to dataset with `ds_train = ds_train.map(augment)`
  - `for x, y in ds_train:`
- **Method 2 -** ImageDataGenerator and flow_from_directory
  - Quite nice if you want to apply a lot of random augmentations to the dataset, it has lots of options
  - `from tensorflow.keras.preprocessing.image import ImageDataGenerator`
  - Using **ImageDataGenerator** allows you to use augmentations, you then call `flow_from_directory` on it

DataLoader

- This is the method used by **pytorch**
- Have a folder with all of the data
- Have a csv file with name:label
- Create a new class that inherits from **Dataset**
  - `from torch.utils.data import Dataset`