# Dogs_and_Cats_reformat

February 12, 2023

[1]:
```
import os
import re

from shutil import copyfile

Debug=True

# Change DATA_DIR_ROOT to wherever you want
# - The raw input data should be in subdirectory train_data_dir
# - The processed data will wind up in subdirectory out_data_dir, with␣
 ↪subdirectories
# -- out_train_dir and out_valid_dir
# --- each of these out directories will have subdirectories "dogs" and "cats"
DATA_DIR_ROOT="/tmp"

train_data_dir= os.path.join(DATA_DIR_ROOT, "train")

out_data_dir= os.path.join(DATA_DIR_ROOT, "out")
out_train_dir= os.path.join(out_data_dir, "train")
out_valid_dir= os.path.join(out_data_dir, "validation")

(cats_train_dir, dogs_train_dir) = [ os.path.join(out_train_dir, label) for␣
 ↪label in ["cats", "dogs"] ]
(cats_valid_dir, dogs_valid_dir) = [ os.path.join(out_valid_dir, label) for␣
 ↪label in ["cats", "dogs"] ]

# Create out directory tree as needed
os.makedirs(out_data_dir, exist_ok=True)
os.makedirs(cats_train_dir, exist_ok=True)
os.makedirs(dogs_train_dir, exist_ok=True)
os.makedirs(cats_valid_dir, exist_ok=True)
os.makedirs(dogs_valid_dir, exist_ok=True)
```

[2]:
```
# Walk the file system tree rooted at train_data_dir
w = os.walk(train_data_dir)

# walk returns a *generator*.  We need only the top level of the tree
```

```
top_path, top_dirs, top_files = list(w)[0]
```

[3]:
```
# Iterate thru files in top directory
for f in top_files:
    # Parse the file name into animal "label" and integer "idx"
    m = re.search(r"^(.*)\.([0-9]+)\.jpg",f)
    label, idx = m.groups()

    # Convert idx from string to integer
    # Output directory is plural of label
    idx = int(idx)
    out_label = label + "s"

    # n.b., file index starts a 0, not 1

    # First 1000 files are for training
    if idx < 1000:
        # Train
        copyfile( os.path.join(top_path,f), os.path.join(out_train_dir,␣
↪out_label, f))
        if Debug and idx < 5:
            print("Found {t:s} {l:s} #{n:d}".format(t="train", l=label, n=idx))

    # Files 1001 - 1400 are for validation
    elif idx < 1400:
        # Validation
        copyfile( os.path.join(top_path,f), os.path.join(out_valid_dir,␣
↪out_label, f))
        if Debug and idx < 1005:
            print("Found {t:s} {l:s} #{n:d}".format(t="validation", l=label,␣
↪n=idx))
    else:
        pass
```

```
Found validation cat #1002
Found train cat #2
Found validation cat #1003
Found validation cat #1004
Found train dog #3
Found train cat #4
Found validation dog #1000
Found train cat #1
Found train dog #0
Found train dog #2
Found validation dog #1002
```

```
Found train cat #0
Found train dog #4
Found validation cat #1001
Found validation dog #1004
Found validation dog #1001
Found train cat #3
Found validation dog #1003
Found validation cat #1000
Found train dog #1
```