

▼ Imports and set vars

```

1 !unzip "/content/drive/My Drive/dog breeds/archive.zip" -d "/content/dog breeds"

1 !pip install -qU fastcore fastai

|████████████████████████████████████████████████████████████████████████████████| 53 kB 1.3 MB/s
|████████████████████████████████████████████████████████████████████████████████| 188 kB 7.6 MB/s

1 from tqdm import tqdm
2 import xml.etree.ElementTree as ET

1 from fastai.vision.all import *
2 path = "/content/dog breeds"
3 Path.BASE_PATH = path

1 dog_images_path = glob.glob(f'{path}/images/Images/*')
2 dog_images = glob.glob(f'{path}/images/Images/**')
3 annotations = "/content/drive/MyDrive/dog breeds/standford dogs annotations.json"

```

▼ Set data loaders

```

1 imgs, lbl_bbox = get_annotations(annotations)

1 imgs[0]

' n02097658_595 '

1 lbl_bbox[0]

([[81, 6, 378, 355]], ['silky_terrier'])

1 append_extension = lambda images: [img + ".jpg" for img in images]

1 img2bbox = dict(zip(append_extension(imgs), lbl_bbox))

1 {k: img2bbox[k] for k in list(img2bbox)[:1]}

{'n02097658_595.jpg': ([[81, 6, 378, 355]], ['silky_terrier'])}

1 get_image_path = lambda class_name: next(x for x in dog_images_path if x.endswit

```

```
1 getters = [lambda x: f"{get_image_path(img2bbox[x][1][0])}/{x}", # get file path
2             lambda x: img2bbox[x][0], # get BB
3             lambda x: img2bbox[x][1]] # get class name

1 item_tfms = [Resize(224)]
2 batch_tfms = [FlipItem(),
3               RandomResizedCrop(128, min_scale=0.35),
4               IntToFloatTensor(),
5               Normalize.from_stats(*imagenet_stats)]

1 def get_train_imgs(noop): return append_extension(imgs)

1 db = DataBlock(blocks=(ImageBlock, BBoxBlock, BBoxLblBlock),
2                 splitter=RandomSplitter(),
3                 get_items=get_train_imgs,
4                 getters=getters,
5                 item_tfms=item_tfms,
6                 batch_tfms=batch_tfms,
7                 n_inp=1)

1 db.summary(dog_images)

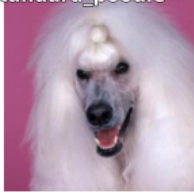
1 set_seed(2*33)

1 dls = db.dataloaders(dog_images)

1 dls.c = 120

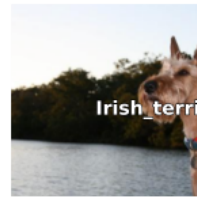
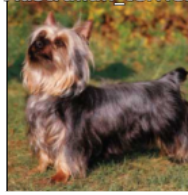
1 dls.show_batch(max_n=20)
```

standard_poodle



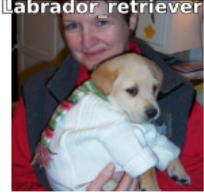
schipperke

Australian_terrier



Irish_terrier

Labrador_retriever



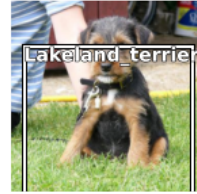
English_springer



Irish_wolfhound



Lhasa



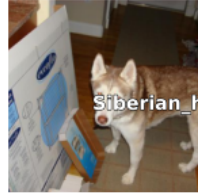
Lakeland_terrier



curly-coated_retriever



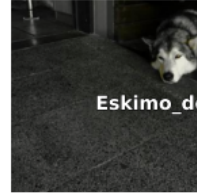
beagle



Siberian_husky



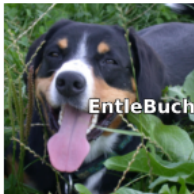
keeshond



Eskimo_dog



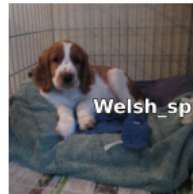
papillon



Entlebucher



dhole



Welsh_springer



giant_schnauzer



toy_poodle

