from duckduckgo\_search import DDGS

from fastcore.all import \*

from fastdownload import download\_url

from fastai.vision.all import \*

def search\_images(term, max\_images=200): return L(DDGS().images(term, max\_results=max\_images)).itemgot('image')

def main():

searches = 'forest','bird'

path = Path('bird\_or\_not')

from time import sleep

for o in searches:

    dest = (path/o)

    dest.mkdir(exist\_ok=True, parents=True)

    download\_images(dest, urls=search\_images(f'{o} photo'))

    sleep(10)  # Pause between searches to avoid over-loading server

    download\_images(dest, urls=search\_images(f'{o} sun photo'))

    sleep(10)

    download\_images(dest, urls=search\_images(f'{o} shade photo'))

    sleep(10)

    resize\_images(path/o, max\_size=400, dest=path/o)

failed = verify\_images(get\_image\_files(path))

failed.map(Path.unlink)

dls = DataBlock(

    blocks=(ImageBlock, CategoryBlock),

    get\_items=get\_image\_files,

    splitter=RandomSplitter(valid\_pct=0.2, seed=42),

    get\_y=parent\_label,

    item\_tfms=[Resize(192, method='squish')]

).dataloaders(path)

dls.show\_batch(max\_n=6)

learn = vision\_learner(dls, resnet18, metrics=error\_rate)

learn.fine\_tune(3)