Importing fastbook and its setups

```
#hide
! [ -e /content ] && pip install -Uqq fastbook
import fastbook
fastbook.setup_book()
```

```
Mounted at /content/gdrive
```

Loading Grizzly bear picture

```
from fastbook import *
urls = search_images_ddg('grizzly bear', max_images=150
len(urls),urls[0]
```

```
(150,
 'http://www.tweedsmuirparklodge.com/assets/Uploa
```

```
dest = 'images/bear.jpg'
download_url(urls[0], dest)

im = Image.open(dest)
im.thumbnail((256,256))
im
```

103.11%

[155648/150956 00:00<00:00]

Saving all the pics in different categories ie bear_types

```
path = Path('bears')
path.mkdir(exist_ok=True)

bear_types = 'grizzly','black','teddy'

for e in bear_types:
  dest = path/e
  dest.mkdir(exist_ok=True)
  bear_images_urls = search_images_ddg(f'{e} bear')
  download_images(dest, urls = bear_images_urls)
```

71qVVUAINNL._AC_SL1000_.jpg •••

```
fns = get_image_files(path)
fns
```

```
(#572) [Path('bears/grizzly/00000053.jpg'),Path('b
```

Acessiing and deleting the saved pics

```
failed = verify_images(fns)
failed
```

```
(#3) [Path('bears/grizzly/00000110.jpg'),Path('bea
```

```
failed.map(Path.unlink);
```

Setting up with data loaders

we have the ImageBlock being the indedependent variable and the dependent variable beng the CategoryBlock and also the *validation percentage set is 0.2*

```
bears = DataBlock(
    blocks=(ImageBlock, CategoryBlock),
    get_items=get_image_files,
    splitter=RandomSplitter(valid_pct=0.2, seed=42),
    get_y=parent_label,
    item_tfms=Resize(128))
```

```
dls = bears.dataloaders(path)
```

```
dls.valid.show_batch(max_n=4, nrows=1)
```


```
bears = bears.new(item_tfms=Resize(128, ResizeMethod.Sq
dls = bears.dataloaders(path)
dls.valid.show_batch(max_n=4, nrows=1)
```

Training my model_bear classiffier (And using it to train our model)

```
bears = bears.new(
    item_tfms=RandomResizedCrop(224, min_scale=0.5),
    batch_tfms=aug_transforms())
dls = bears.dataloaders(path)
```

Create our learner and fine tune(CNN learner)

```
learn = vision_learner(dls, resnet18, metrics=error_rat
learn.fine_tune(4)
```

```
Downloading: "https://download.pytorch.org/models/
100%  44.7M/44.7M [00:00<00:00, 80.5MB/s]
```

epoch	train_loss	valid_loss	error_rate	time
0	0.861993	0.076917	0.026549	00:27

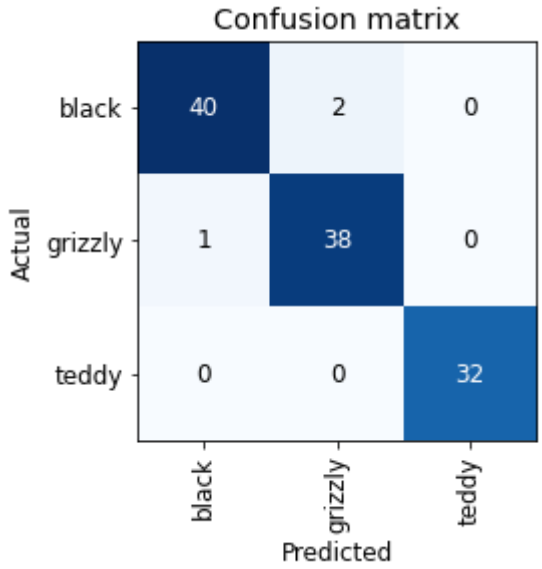
```
/usr/local/lib/python3.7/dist-packages/PIL/Image.p
  "Palette images with Transparency expressed in b
```

epoch	train_loss	valid_loss	error_rate	time
0	0.099805	0.069206	0.017699	00:28
1	0.074326	0.078801	0.017699	00:28
2	0.076075	0.107011	0.026549	00:28
3	0.065536	0.121330	0.026549	00:27

```
/usr/local/lib/python3.7/dist-packages/PIL/Image.p
  "Palette images with Transparency expressed in b
/usr/local/lib/python3.7/dist-packages/PIL/Image.p
  "Palette images with Transparency expressed in b
/usr/local/lib/python3.7/dist-packages/PIL/Image.p
  "Palette images with Transparency expressed in b
```

Comfusion matrix

```
interp = ClassificationInterpretation.from_learner(lear
interp.plot_confusion_matrix()
```



Top losses(Predict-Actual-loss-Probality)

```
interp.plot_top_losses(5, nrows=1)
```

Installing fast AI and useful libraries

```
!pip install fastai
```

```
Requirement already satisfied: fastai in /usr/loca
Requirement already satisfied: fastcore<1.5,>=1.3
Requirement already satisfied: matplotlib in /usr/
Requirement already satisfied: scipy in /usr/local
Requirement already satisfied: torchvision>=0.8.2
Requirement already satisfied: requests in /usr/lo
Requirement already satisfied: pyyaml in /usr/loca
Requirement already satisfied: pandas in /usr/loca
Requirement already satisfied: scikit-learn in /us
Requirement already satisfied: fastprogress>=0.2.4
Requirement already satisfied: pillow>6.0.0 in /us
Requirement already satisfied: torch<1.12,>=1.7.0
```

```
Requirement already satisfied: packaging in /usr/l
Requirement already satisfied: pip in /usr/local/l
Requirement already satisfied: spacy<4 in /usr/loc
Requirement already satisfied: fastdownload<2,>=0
Requirement already satisfied: setuptools in /usr/
Requirement already satisfied: murmurhash<1.1.0,>=
Requirement already satisfied: cymem<2.1.0,>=2.0.2
Requirement already satisfied: numpy>=1.15.0 in /u
Requirement already satisfied: plac<1.2.0,>=0.9.6
Requirement already satisfied: thinc==7.4.0 in /us
Requirement already satisfied: tqdm<5.0.0,>=4.38.0
Requirement already satisfied: srsly<1.1.0,>=1.0.2
Requirement already satisfied: preshed<3.1.0,>=3.0
Requirement already satisfied: wasabi<1.1.0,>=0.4
Requirement already satisfied: catalogue<1.1.0,>=0
Requirement already satisfied: blis<0.5.0,>=0.4.0
Requirement already satisfied: importlib-metadata<
Requirement already satisfied: zipp>=0.5 in /usr/l
Requirement already satisfied: typing-extensions>=
Requirement already satisfied: urllib3!=1.25.0,!=1
Requirement already satisfied: idna<3,>=2.5 in /us
Requirement already satisfied: certifi>=2017.4.17
Requirement already satisfied: chardet<4,>=3.0.2 i
Requirement already satisfied: pyparsing!=2.0.4,!=
Requirement already satisfied: kiwisolver>=1.0.1 i
Requirement already satisfied: cycler>=0.10 in /us
Requirement already satisfied: python-dateutil>=2
Requirement already satisfied: six>=1.5 in /usr/lo
Requirement already satisfied: pytz>=2017.3 in /us
Requirement already satisfied: joblib>=0.11 in /us
Requirement already satisfied: threadpoolctl>=2.0
```

```
from fastai.vision.widgets import *


#hide_output
cleaner = ImageClassifierCleaner(learn)
cleaner
```

```
/usr/local/lib/python3.7/dist-packages/PIL/Image.p
  "Palette images with Transparency expressed in b
```

black

Train

<Keep> <Keep>

Predicting and deploying

Turning Your Model into an Online Application

Using the Model for Inference

```
learn.export()
```

```
path = Path()
path.ls(file_exts='.pkl')
```

```
(#1) [Path('export.pkl')]
```

```
learn_inf = load_learner(path/'export.pkl')
```

predicting if the category of an animal in a pic

```
learn_inf.predict('/content/71qVVUAINNL._AC_SL1000_.jpg
```

```
('teddy', TensorBase(2), TensorBase([2.2119e-09,
```

**Our targets which will always be predicted*

```
learn_inf.dls.vocab
```

```
['black', 'grizzly', 'teddy']
```

✓ 0s completed at 4:17 PM