```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=

Enter your authorization code:
··········
Mounted at /content/gdrive
```

```
!mkdir -p data
!mkdir -p data/orientation
!cp /content/gdrive/"My Drive"/Projects/kuiken-ulcers/physionet-classifier/orientatio
!unzip data/orientation_dataset.zip -d data/orientation >/dev/null 2>&1
```

```
from pathlib import Path
from fastai.vision import *
from fastai.metrics import *
```

```
data_path = Path("data/orientation")
```

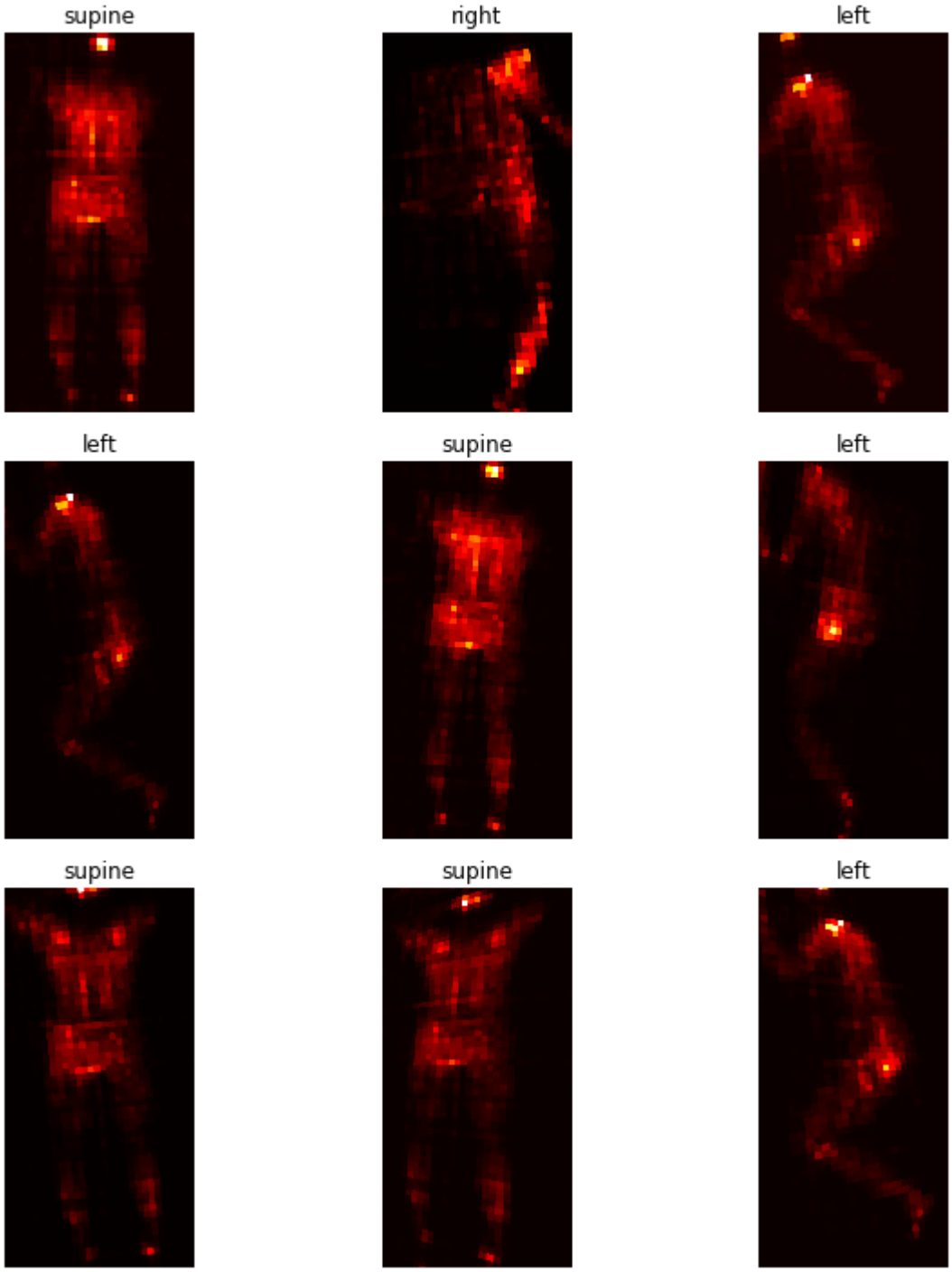
```
data_path.ls()
```

```
[PosixPath('data/orientation/left'),
 PosixPath('data/orientation/supine'),
 PosixPath('data/orientation/right')]
```

```
batch_size = 24
```

```
data = ImageDataBunch.from_folder(
    path=data_path,
    valid_pct=0.2,
    ds_tfms=get_transforms(do_flip=False),
    bs=batch_size
)
```

```
data.show_batch(rows=3, figsize=(10,10))
```




```
print(data.classes)
```

```
['left', 'right', 'supine']
```

```
learn = cnn_learner(data, models.resnet34, metrics=accuracy)
```

```
Downloading: "https://download.pytorch.org/models/resnet34-333f7ec4.pth" to /root
100%|██████████| 83.3M/83.3M [00:03<00:00, 27.6MB/s]
```

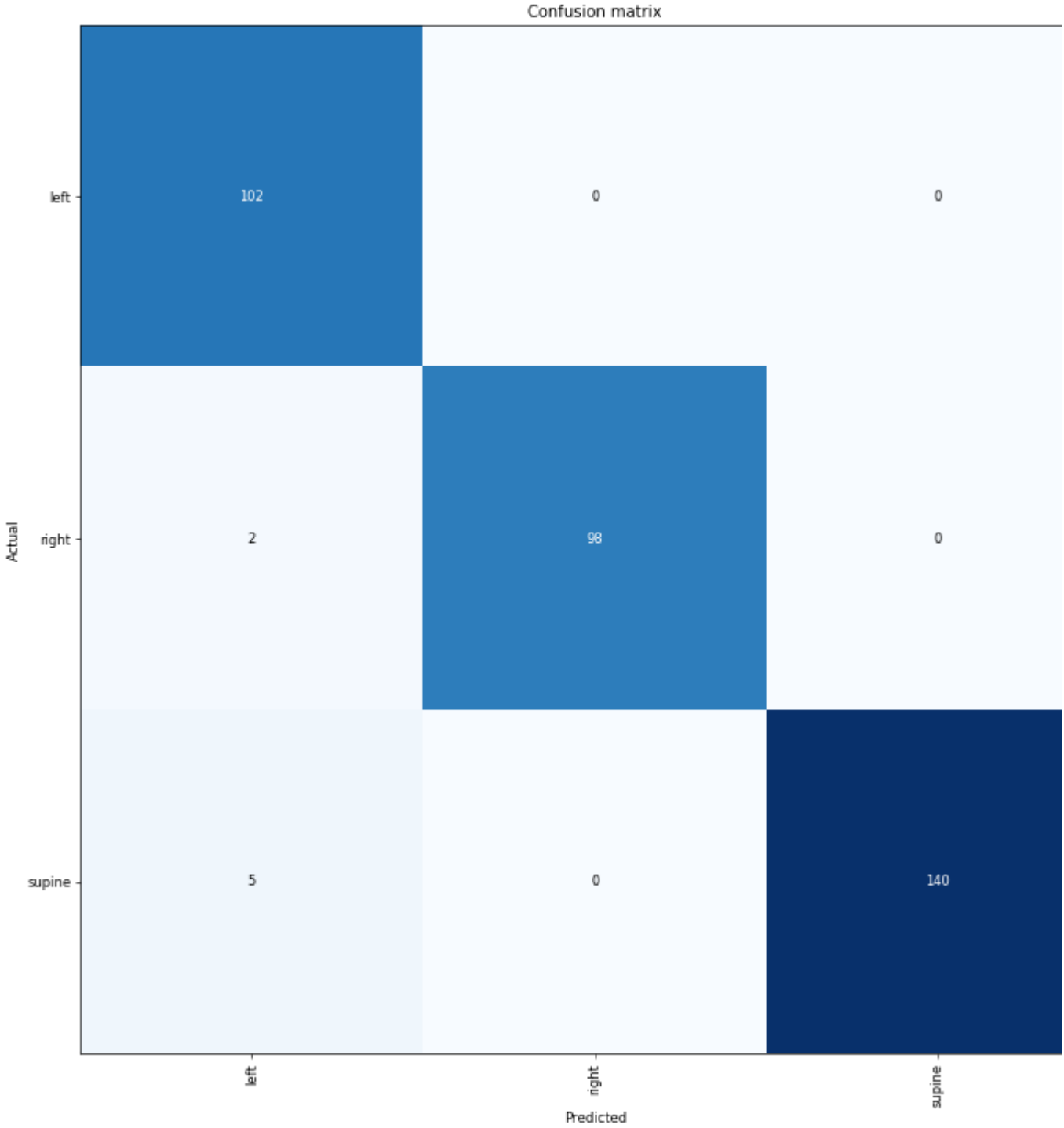
```
learn.fit_one_cycle(2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.681993	0.099618	0.985591	01:26
1	0.299381	0.083538	0.988473	01:20
2	0.161027	0.166224	0.979827	01:23
3	0.132460	0.081760	0.979827	01:23

```
learn.recorder.plot_losses()
```

```
learn.save("initial-training")
```

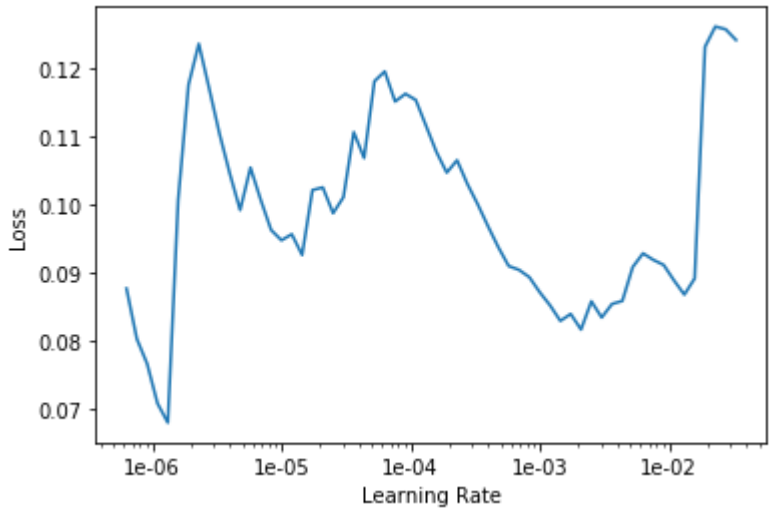
```
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix(figsize=(7,7), dpi=60)
```




```
learn.lr_find()
```

```
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```

```
learn.recorder.plot()
```



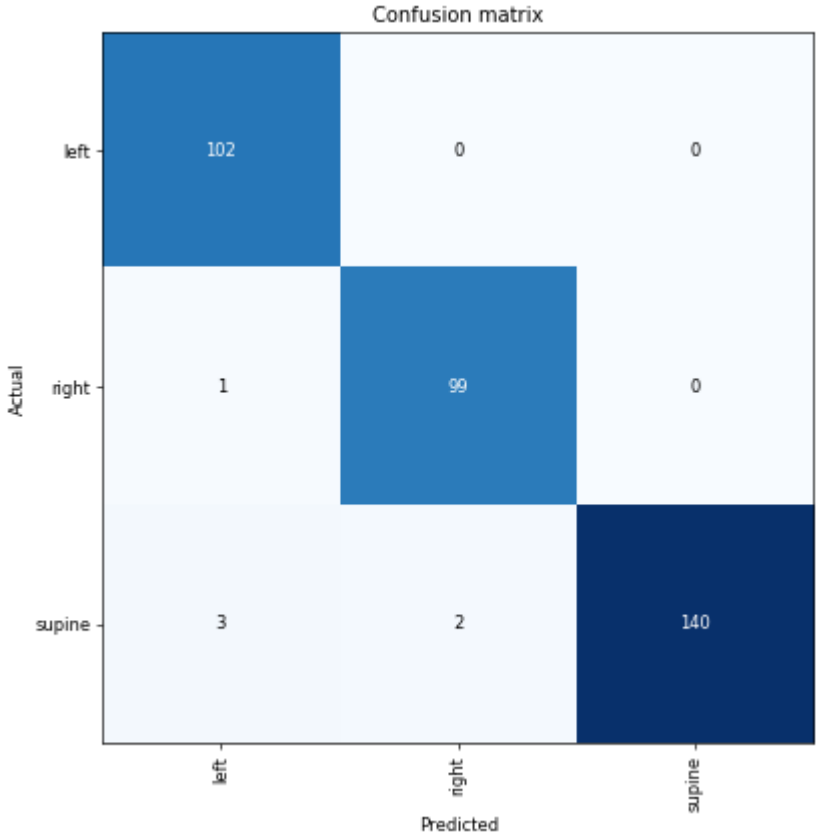

```
learn.unfreeze()
learn.fit_one_cycle(2, max_lr=slice(1e-4,1e-3))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.291824	1.597319	0.804035	01:42
1	0.171718	0.155558	0.982709	01:40

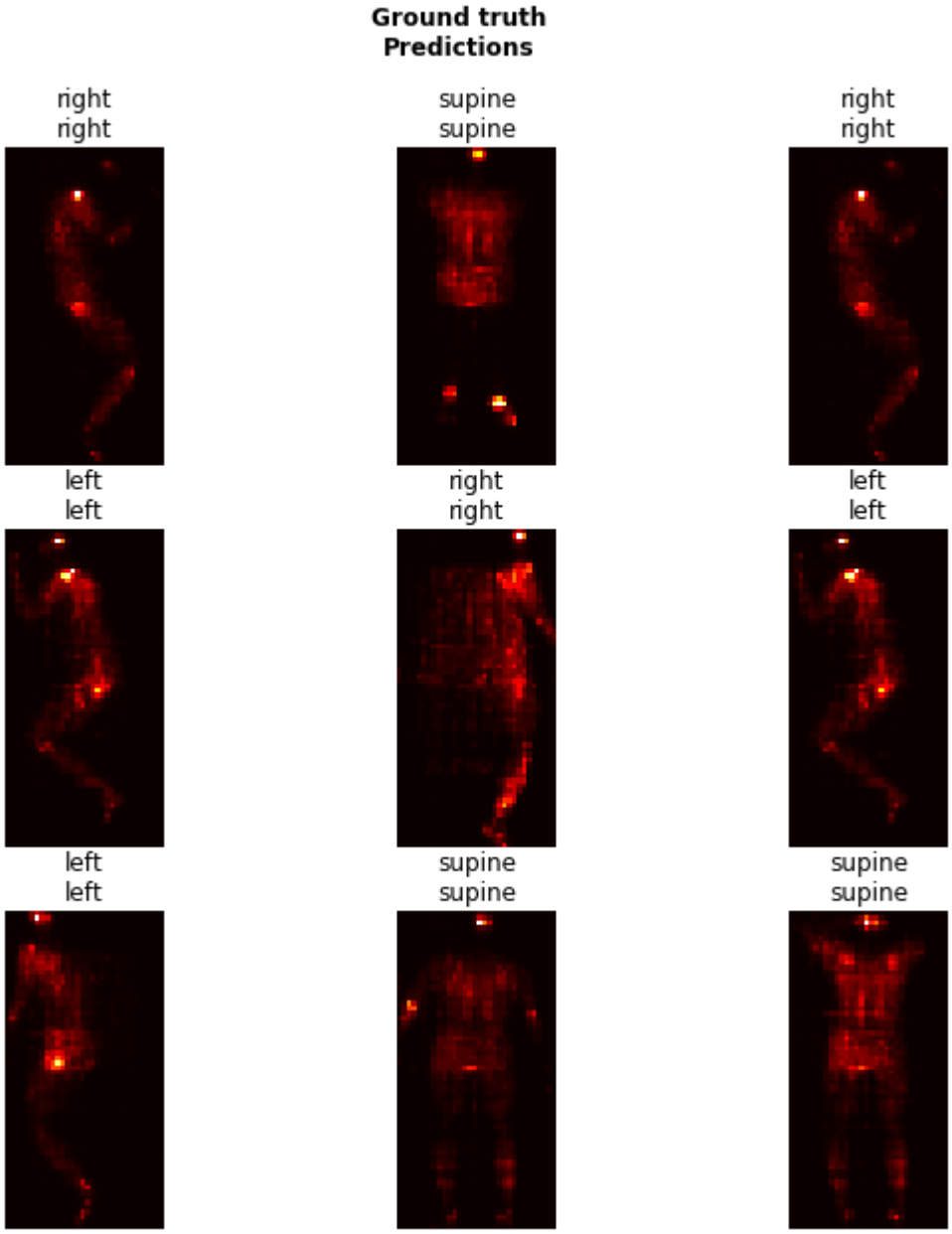
```
learn.recorder.plot_losses()
```

```
learn.save("second-training")
```

```
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix(figsize=(7,7), dpi=60)
```




```
learn.show_results(rows=3, figsize=(10,10))
```


Ground truth
Predictions
right
right
supine
supine
right
right
left
left
right
right
left
left
left
left
supine
supine
supine
supine