

```
In [1]: from fastai.vision import *
```

```
In [2]: torch.__version__
```

```
Out[2]: '1.1.0'
```

```
In [3]: classes = ['Anger', 'Disgust', 'Surprise', 'Sadness', 'Happiness',
```

```
In [4]: data = ImageDataBunch.single_from_classes('', classes, ds_tfms=get_
```

```
In [5]: learner = cnn_learner(data, models.resnet34)
```

```
In [6]: learner.load('gokul-sentiment-stage-5n')
track_running_stats=True)
  (7): Dropout(p=0.5)
  (8): Linear(in_features=512, out_features=8, bias=True)
)
), opt_func=functools.partial(<class 'torch.optim.adam.Adam'>, b
etas=(0.9, 0.99)), loss_func=FlattenedLoss of CrossEntropyLoss()
, metrics=[], true_wd=True, bn_wd=True, wd=0.01, train_bn=True,
path=PosixPath('.'), model_dir='models', callback_fns=[functools
.partial(<class 'fastai.basic_train.Recorder'>, add_time=True, s
ilent=False)], callbacks=[], layer_groups=[Sequential(
  (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=
(3, 3), bias=False)
  (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, tra
ck_running_stats=True)
  (2): ReLU(inplace)
  (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
  (4): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding
=(1, 1), bias=False)
  (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, tra
```

```
In [7]: path = Path('data/faces')
dest = path
dest.mkdir(parents=True, exist_ok=True)
```

```
In [8]: pwd
```

```
Out[8]: '/home/streicher/aws-lambda'
```

```
In [9]: img = open_image(path/'face.jpeg')
img
```

```
Out[9]:
```



```
In [11]: pred_class, pred_idx, losses = learner.predict(img)
print(pred_class)
Surprise
```

```
In [12]: analysis = {
    "predictions": dict(sorted(
        zip(learner.data.classes, map(float, losses)),
        key=lambda p: p[1],
        reverse=True
    ))
}
analysis
```

```
Out[12]: {'predictions': {'Surprise': 0.8788825869560242,
    'Fear': 0.07538612931966782,
    'Neutral': 0.029466265812516212,
    'Anger': 0.011372542940080166,
    'Sadness': 0.00285532814450562,
    'Happiness': 0.0009696771157905459,
    'Disgust': 0.0008966695750132203,
    'Contempt': 0.0001708116615191102}}
```

```
In [13]: path_img = Path('.')
# export model to TorchScript format
trace_input = torch.ones(1,3,299,299).cuda()
jit_model = torch.jit.trace(learner.model.float(), trace_input)
model_file='resnet34_jit.pth'
output_path = str(path_img/f'models/{model_file}')
torch.jit.save(jit_model, output_path)
# export classes text file
save_texts(path_img/'models/classes.txt', classes)
tar_file=path_img/'models/model.tar.gz'
classes_file='classes.txt'
# create a tarfile with the exported model and classes text file
with tarfile.open(tar_file, 'w:gz') as f:
    f.add(path_img/f'models/{model_file}', arcname=model_file)
    f.add(path_img/f'models/{classes_file}', arcname=classes_file)
```

```
In [14]: import boto3  
s3 = boto3.resource('s3')
```

```
In [15]: s3.meta.client.upload_file('models/model.tar.gz', 'streicher-fastai')
```