

```
In [1]: %reload_ext autoreload
        %autoreload 2
        %matplotlib inline
```

```
In [2]: from fastai.imports import *
        from fastai.transforms import *
        from fastai.conv_learner import *
        from fastai.model import *
        from fastai.dataset import *
        from fastai.sgdr import *
        from fastai.plots import *
```

```
In [3]: PATH = "data/elephantmouse/"
```

```
In [4]: sz=224
```

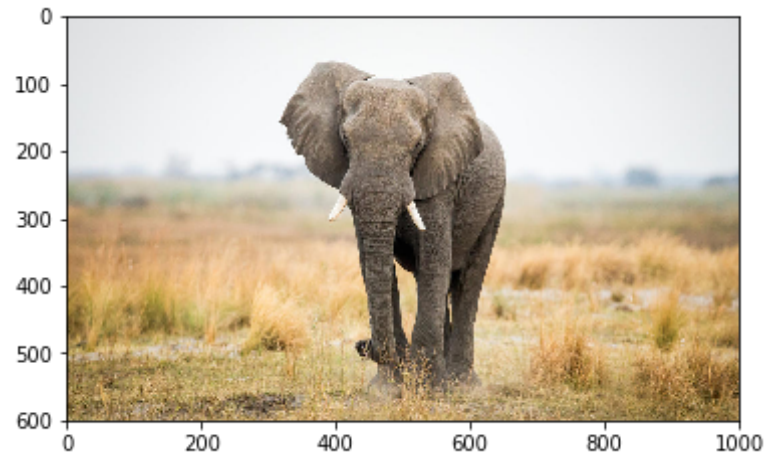
```
In [5]: !ls {PATH}
        models tmp train valid
```

```
In [6]: !ls {PATH}valid
        elephant mouse
```

```
In [7]: files = !ls {PATH}valid/elephant | head
        files
```

```
Out[7]: ['e1.jpg', 'e2.jpg', 'e3.jpg', 'e4.jpg', 'e5.jpg', 'e6.jpg', 'e7.jpg']
```

```
In [8]: img = plt.imread(f'{PATH}valid/elephant/{files[0]}')
        plt.imshow(img);
```



```
In [9]: img.shape
```

```
Out[9]: (600, 1000, 3)
```

```
In [10]: img[:4,:4]
```

```
Out[10]: array([[222, 223, 225],
                [222, 223, 225],
                [222, 223, 225],
                [222, 223, 225]],

               [[222, 223, 225],
                [222, 223, 225],
                [222, 223, 225],
                [222, 223, 225]],

               [[222, 223, 225],
                [222, 223, 225],
                [222, 223, 225],
                [222, 223, 225]],

               [[222, 223, 225],
                [222, 223, 225]]]
```

```
[222, 223, 225],  
[222, 223, 225]]], dtype=uint8)
```

```
In [11]: arch=resnet34  
data = ImageClassifierData.from_paths(PATH, tfms=tfms_from_model(arch,  
sz))  
learn = ConvLearner.pretrained(arch, data, precompute=True)  
learn.fit(0.01, 3)
```

```
[ 0.      1.66725  1.6991   0.      ]  
[ 1.      1.49811  1.11781  0.28571]  
[ 2.      1.3445   0.68463  0.71429]
```

```
In [12]: data.val_y
```

```
Out[12]: array([1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2])
```

```
In [13]: data.classes
```

```
Out[13]: ['.ipynb_checkpoints', 'elephant', 'mouse']
```

```
In [14]: rm -rf .ipynb_checkpoints
```

```
In [15]: data.classes
```

```
Out[15]: ['.ipynb_checkpoints', 'elephant', 'mouse']
```

```
In [16]: log_preds = learn.predict()  
log_preds.shape
```

```
Out[16]: (14, 3)
```

```
In [17]: log_preds[:10]
```

```
Out[17]: array([[ -0.59457, -0.90222, -3.15753],  
                [-0.98265, -0.53246, -3.25648],
```

```
[-1.14292, -0.50566, -2.55097],  
[-1.18796, -0.46048, -2.74611],  
[-1.22466, -0.43365, -2.8472 ],  
[-0.66289, -0.84235, -2.91986],  
[-0.38033, -1.20462, -4.10082],  
[-0.45607, -2.95148, -1.15845],  
[-1.19844, -2.23865, -0.5247 ],  
[-1.04438, -3.82513, -0.46796]], dtype=float32)
```