Assessment 1: I can train and deploy a neural network

At this point, you've worked through a full deep learning workflow. You've loaded a dataset, trained a model, and deployed your model into a simple application. Validate your learning by attempting to replicate that workflow with a new problem.

We've included a dataset which consists of two classes:

1) Face: Contains images which include the face of a whale
2) Not Face: Contains images which do not include the face of a whale.

The dataset is located at `/dli/data/whale/data/train` .

Your challenge is:

1) Use DIGITS (/digits) to train a model to identify *new* whale faces with an accuracy of more than 80%.

2) Deploy your model by modifying and saving the python application submission.py (../../../../edit/tasks/task-assessment/task/submission.py) to return the word "whale" if the image contains a whale's face and "not whale" if the image does not.

Resources:

1) Train a model (../../task1/task/Train%20a%20Model.ipynb)
2) New Data as a goal (../../task2/task/New%20Data%20as%20a%20Goal.ipynb)
3) Deployment (../../task3/task/Deployment.ipynb)

Suggestions:

- Use empty code blocks to find out any informantion necessary to solve this problem: eg: `!ls [directorypath] prints the files in a given directory`
- Executing the first two cells below will run your python script with test images, the first should return "whale" and the second should return "not whale"

Start in DIGITS (/digits/).

In [1]:

```
!python submission.py '/dli/data/whale/data/train/face/w_1.jpg'  #This should return "whale" at the very bottom
```

```
  File "submission.py", line 22
    prediction = net.##REPLACE WITH THE FUNCTION THAT RETURNS THE OUTPUT OF THE NETWORK##([##REPLACE WITH THE INPU
T TO THE FUNCTION##])

                     ^
SyntaxError: invalid syntax
```

In [ ]:

```
!python submission.py '/dli/data/whale/data/train/not_face/w_1.jpg'  #This should return "not whale" at the very bo
```

In [4]:

```
!ls /dli/data/whale/data/train/not_face
```

```
w_1.jpg     w_2489.jpg  w_3902.jpg  w_5290.jpg  w_6661.jpg  w_8155.jpg
w_100.jpg   w_249.jpg   w_3904.jpg  w_5293.jpg  w_6664.jpg  w_8156.jpg
w_1000.jpg  w_2490.jpg  w_3908.jpg  w_5294.jpg  w_6665.jpg  w_8157.jpg
w_1003.jpg  w_2491.jpg  w_3909.jpg  w_5296.jpg  w_6666.jpg  w_8159.jpg
w_1004.jpg  w_2492.jpg  w_391.jpg   w_5297.jpg  w_6669.jpg  w_8160.jpg
w_1005.jpg  w_2493.jpg  w_3910.jpg  w_5298.jpg  w_6673.jpg  w_8161.jpg
w_1006.jpg  w_2494.jpg  w_3913.jpg  w_5299.jpg  w_6674.jpg  w_8164.jpg
w_1007.jpg  w_2495.jpg  w_3916.jpg  w_530.jpg   w_6675.jpg  w_8170.jpg
w_1011.jpg  w_2496.jpg  w_3917.jpg  w_5302.jpg  w_6676.jpg  w_8171.jpg
w_1012.jpg  w_2499.jpg  w_3918.jpg  w_5303.jpg  w_668.jpg   w_8172.jpg
w_1014.jpg  w_2502.jpg  w_3919.jpg  w_5304.jpg  w_6680.jpg  w_8175.jpg
w_1016.jpg  w_2505.jpg  w_3920.jpg  w_5307.jpg  w_6682.jpg  w_8177.jpg
w_1017.jpg  w_251.jpg   w_3921.jpg  w_5308.jpg  w_6685.jpg  w_818.jpg
w_1024.jpg  w_2510.jpg  w_3925.jpg  w_5309.jpg  w_6687.jpg  w_8180.jpg
w_1026.jpg  w_2518.jpg  w_393.jpg   w_531.jpg   w_6689.jpg  w_8181.jpg
w_1029.jpg  w_2519.jpg  w_3931.jpg  w_5310.jpg  w_669.jpg   w_8182.jpg
w_1031.jpg  w_2520.jpg  w_3932.jpg  w_5312.jpg  w_6690.jpg  w_8184.jpg
w_1033.jpg  w_2524.jpg  w_3933.jpg  w_5313.jpg  w_6694.jpg  w_8185.jpg
w_1035.jpg  w_2525.jpg  w_3934.jpg  w_5318.jpg  w_6695.jpg  w_819.jpg
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: