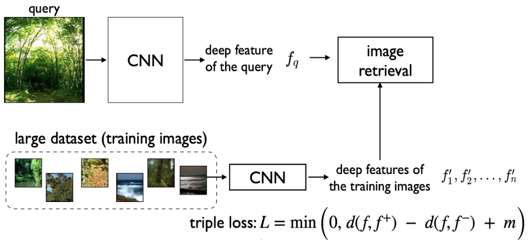
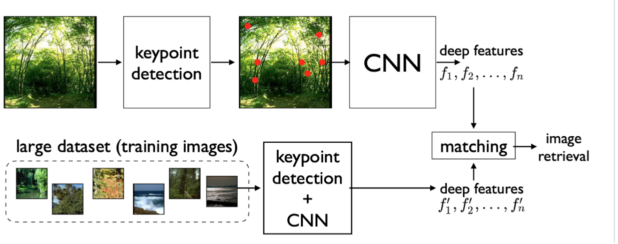
Final Project Status Update
Qingxiao Yuan

(1) Problem statement

I will do the scene binary classification problem. Specifically, if we are given an image of the scene (256 x 256), determine whether it is coast or forest by using what we learnt in this course.

(2) Approach


query
CNN
deep feature of the query
f_q
image retrieval
large dataset (training images)
CNN
deep features of the training images
$f'_1, f'_2, \dots, f'_n$
triple loss: $L = \min\left(0, d(f,f^+) - d(f,f^-) + m\right)$



1.Use the cv2 library to implement this task. First, detect the 15 SIFT keypoints in the image, as we learnt in class. Then, extract the (32 x 32) patches from each keypoint.

2. Then, compute a 128-dimensional deep feature by CNN & Triple Loss

The library I use is pytorch, torch.nn. Implementation details is following:

Learning rate: 0.3; Epoch: 80 ; CNN Layer: 2;Optimizer : Stochastic gradient descent (sgd), reducing the triple loss to determine the image

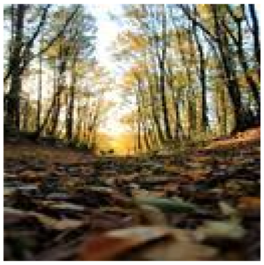
3. Using the Keypoint and CNN, One-One Matching to compute a 128-dimensional deep feature

Library: scipy.optimize.linear_sum_assignment, Determine the image by similarity.

(3) Datasets

https://drive.google.com/drive/folders/1fjCtMBwbhGsbQAq1QI71rmAe8TinTCTy?usp=sharing

25 coast test images, 50 coast train images, 25 forest test images, and 50 forest train images.

In ground_truth.txt: the first part index 0-24 represents the coast and the second part index 25-49 stands for the forest.  *(coast(0), forest(0))*

The main challenge is that the small number of the dataset may lead to overfitting. Additionally, some scenes may be pretty similar or contain each other. For example, there may be a lake in a forest, or a lake with forest shadow, which will increase the test burden.

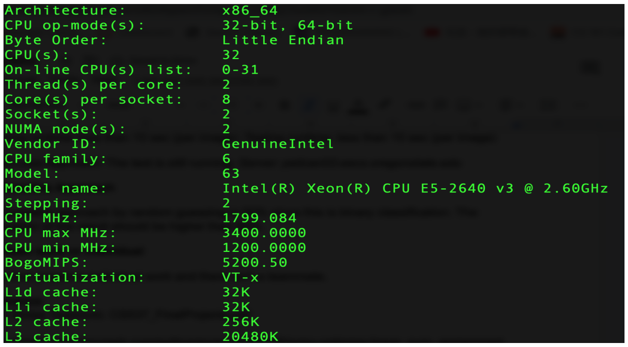
(4) **Preliminary Quantitative Results**

Matching Approach:

Number of images	50
Precise	57%

Training runtime: less than 13 sec (per image) ;Testing runtime: less than 13 sec (per image)

Triple Loss Approach: The test is still running. *Server: pelican03.eecs.oregonstate.edu*



```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Thread(s) per core:    2
Core(s) per socket:    8
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 63
Model name:            Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz
Stepping:              2
CPU MHz:               1799.084
CPU max MHz:           3400.0000
CPU min MHz:           1200.0000
BogoMIPS:              5200.50
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              20480K
```

(5) Baseline Approach

I'll use the intersection over union to evaluate the result. The higher, the better. It is a similarity problem. The baseline of the result should be higher than 60%.

(6) Team work or individual

Individual work.

Reference

Prof. Sinisa Todorovic. CS537_FinalProjects.pdf

Prof. Sinisa Todorovic. CS537_7.pdf

https://www.programcreek.com/python/example/97225/scipy.optimize.linear_sum_assignment