WK5 Report

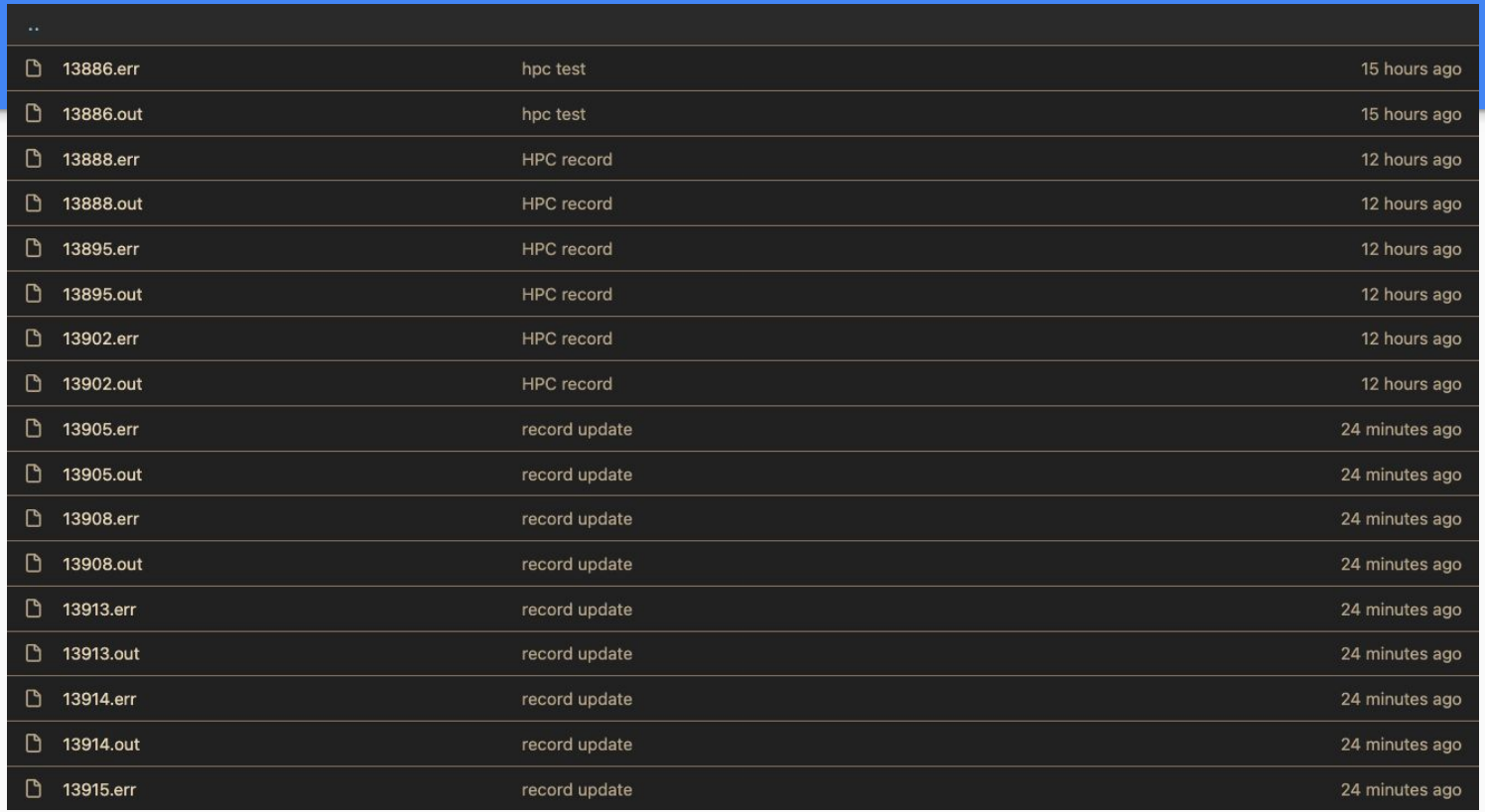
Replicate the paper

Yunting Chiu

Note

1. Now in HPC domain
2. Start to use py script instead of py notebook (Colab)
3. Study PyTorch

Code Debugging via HPC



..		
13886.err	hpc test	15 hours ago
13886.out	hpc test	15 hours ago
13888.err	HPC record	12 hours ago
13888.out	HPC record	12 hours ago
13895.err	HPC record	12 hours ago
13895.out	HPC record	12 hours ago
13902.err	HPC record	12 hours ago
13902.out	HPC record	12 hours ago
13905.err	record update	24 minutes ago
13905.out	record update	24 minutes ago
13908.err	record update	24 minutes ago
13908.out	record update	24 minutes ago
13913.err	record update	24 minutes ago
13913.out	record update	24 minutes ago
13914.err	record update	24 minutes ago
13914.out	record update	24 minutes ago
13915.err	record update	24 minutes ago

Algorithm 1 DeepFake Detection Algorithm

Input : D_{real}, D_{fake} were centered by subtracting the mean of the real training data,

```
def meanSubtraction(arr):
  new_arr = []
  for i in range(len(arr)):
    img = arr[i]
    img = img.astype(np.float32) # convert from integers to floats
    mean = img.mean() # calculate global mean
    img = img - mean # centering of pixels
    #img /= img.std()
    #img = [np.round(img, 2) for i in range(len(arr))]
    new_arr.append(img)
  new_arr = np.array(new_arr, dtype=object)
  return new_arr

img_real = meanSubtraction(img_real)
# print(len(img_real))
# print(img_real[0])
```

Algorithm 1 DeepFake Detection Algorithm

Input : $\mathbf{D}_{real}, \mathbf{D}_{fake}$ were centered by subtracting the mean of the real training data.

(1) Preprocessing and data tensor organization:

$$[\mathbf{U}_{real}, \mathbf{S}_{real}, \mathbf{V}_{real}] \Leftarrow svd(\mathbf{D}_{real})$$

$$[\mathbf{U}_{fake}, \mathbf{S}_{fake}, \mathbf{V}_{fake}] \Leftarrow svd(\mathbf{D}_{fake})$$

$$\mathcal{D}(:, :, 1) = [\mathbf{U}_{real}\mathbf{S}_{real}]$$

$$\mathcal{D}(:, :, 2) = [\mathbf{U}_{fake}\mathbf{S}_{fake}]$$

One of the shape in U_real is (480, 720, 3)

One of the shape in S_real is (480, 3)

One of the shape in V_real is (720, 3, 3)

```
Successfully completed.

Resource usage summary:

    CPU time :                                   250.34 sec.
    Max Memory :                                 77052 MB
    Average Memory :                             33048.55 MB
    Total Requested Memory :                     -
    Delta Memory :                               -
    Max Swap :                                   2834 MB
    Max Processes :                              4
    Max Threads :                                28
    Run time :                                   388 sec.
    Turnaround time :                            388 sec.

The output (if any) follows:

Total image is 7000
U_real is
[array([[[-0.04080221,  0.01320603, -0.04130071],
        [-0.03989985,  0.01289491, -0.04052035],
        [-0.03989985,  0.01289491, -0.04052035],
        ...,
        [-0.04226176,  0.00201746, -0.02599541],
        [-0.04226803, -0.00489948, -0.03500889]
```

Algorithm 1 DeepFake Detection Algorithm

Input : $\mathbf{D}_{\text{real}}, \mathbf{D}_{\text{fake}}$ were centered by subtracting the mean of the real training data,

(1) Preprocessing and data tensor organization:

$[\mathbf{U}_{\text{fake}}, \mathbf{S}_{\text{fake}}, \mathbf{V}_{\text{fake}}] \Leftarrow \text{svd}(\mathbf{D}_{\text{fake}})$

$\mathcal{D}(:, :, 1) = [\mathbf{U}_{\text{real}}\mathbf{S}_{\text{real}}]$

$\mathcal{D}(:, :, 2) = [\mathbf{U}_{\text{fake}}\mathbf{S}_{\text{fake}}]$

One of the shape in U_fake is (480, 640, 3)

One of the shape in S_fake is (480, 3)

One of the shape in V_fake is (480, 3, 3)

```
/app/python3/bin/python3 D_fake.py

------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   157.91 sec.
    Max Memory :                                 77970 MB
    Average Memory :                             25947.52 MB
    Total Requested Memory :                     -
    Delta Memory :                               -
    Max Swap :                                   -
    Max Processes :                              4
    Max Threads :                                28
    Run time :                                   184 sec.
    Turnaround time :                            184 sec.

The output (if any) follows:

Total image is 7000
[array([[[33, 38, 36],
         [33, 38, 36],
         [33, 38, 36],
         ...,
         [ 4,  4,  4],
```

Creates a Tensor from a numpy.ndarray

When an array has three or more than three dimensions, we call it a tensor

PyTorch is a library for Python programs that facilitates building deep learning projects

Credit: https://towardsdatascience.com/what-is-pytorch-a84e4559f0e3

Algorithm 1 DeepFake Detection Algorithm

Input : $\mathbf{D}_{real}, \mathbf{D}_{fake}$ were centered by subtracting the mean of the real training data,

(1) Preprocessing and data tensor organization:

$[\mathbf{U}_{real}, \mathbf{S}_{real}, \mathbf{V}_{real}] \Leftarrow \text{svd}(\mathbf{D}_{real})$

$[\mathbf{U}_{fake}, \mathbf{S}_{fake}, \mathbf{V}_{fake}] \Leftarrow \text{svd}(\mathbf{D}_{fake})$

$\mathcal{D}(:, :, 1) = [\mathbf{U}_{real}\mathbf{S}_{real}]$

$\mathcal{D}(:, :, 2) = [\mathbf{U}_{fake}\mathbf{S}_{fake}]$