

Alternative Method Memory Usage

Filename: train.py

Line #	Mem usage	Increment	Line Contents
202	73.797 MiB	73.797 MiB	@profile
203			def main():
...			
251	98.793 MiB	0.000 MiB	if args.classifyalt:
252	98.793 MiB	0.000 MiB	faces_train =
faces_train.reshape(n_faces, int(faces_train.shape[0]/n_faces), n_pixels)			
253	98.793 MiB	0.000 MiB	target_train =
target_train.reshape(n_faces, int(target_train.shape[0]/n_faces))			
254			
255	98.855 MiB	0.062 MiB	distances = np.zeros((n_faces,
faces_test.shape[0]))			
256	100.180 MiB	0.000 MiB	for i in range(n_faces):
257	100.180 MiB	0.000 MiB	target_pred, distances[i]
= test_model(args.eigen, faces_train[i],			
258	100.180 MiB	1.324 MiB	faces_test,
target_train[i], target_test, args)			
259	100.184 MiB	0.004 MiB	target_pred =
np.argmin(distances, axis=0)			
...			

=====

```

202 73.797 MiB 73.797 MiB @profile
203
...
251 98.793 MiB 0.000 MiB     if args.classifyalt:
252 98.793 MiB 0.000 MiB         faces_train =
faces_train.reshape(n_faces, int(faces_train.shape[0]/n_faces), n_pixels)
253 98.793 MiB 0.000 MiB         target_train =
target_train.reshape(n_faces, int(target_train.shape[0]/n_faces))
254
255 98.855 MiB 0.062 MiB         distances = np.zeros((n_faces,
faces_test.shape[0]))
256 100.180 MiB 0.000 MiB         for i in range(n_faces):
257 100.180 MiB 0.000 MiB             target_pred, distances[i]
= test_model(args.eigen, faces_train[i],
258 100.180 MiB 1.324 MiB                 faces_test,
target_train[i], target_test, args)
259 100.184 MiB 0.004 MiB             target_pred =
np.argmin(distances, axis=0)
...

```