HW 11

Adam Karl

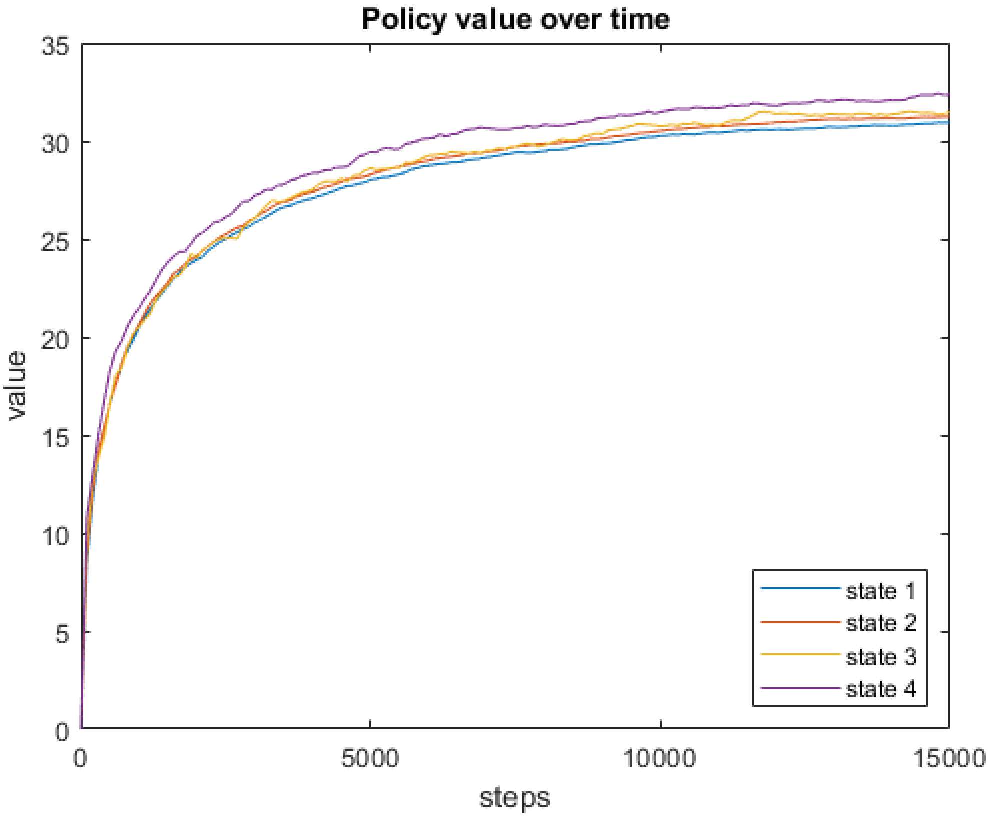
April 24, 2021

1 Reinforcement Learning Agent

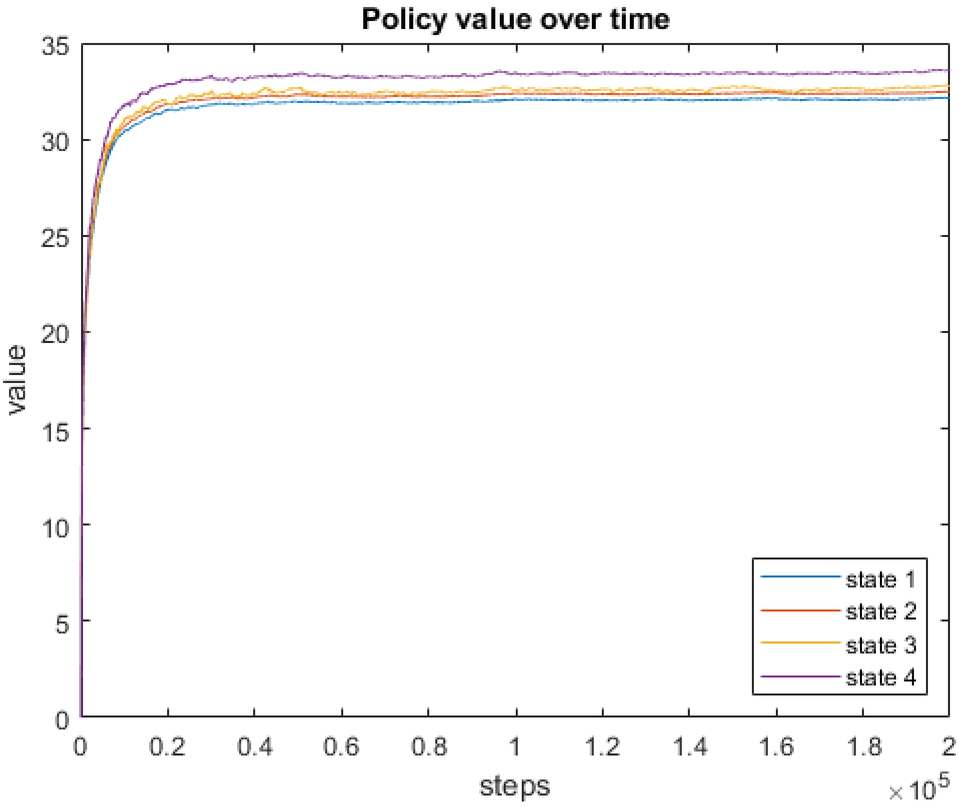
b.

state	action	reward
2	1	1
3	3	2
4	2	3
1	1	1
1	1	2
2	1	1
3	3	0
2	1	1
3	3	0
2	1	1
3	-	-

c.
After 15000 steps, the calculated policy values were $[30.9, 31.2, 31.4, 32.4]^T$.

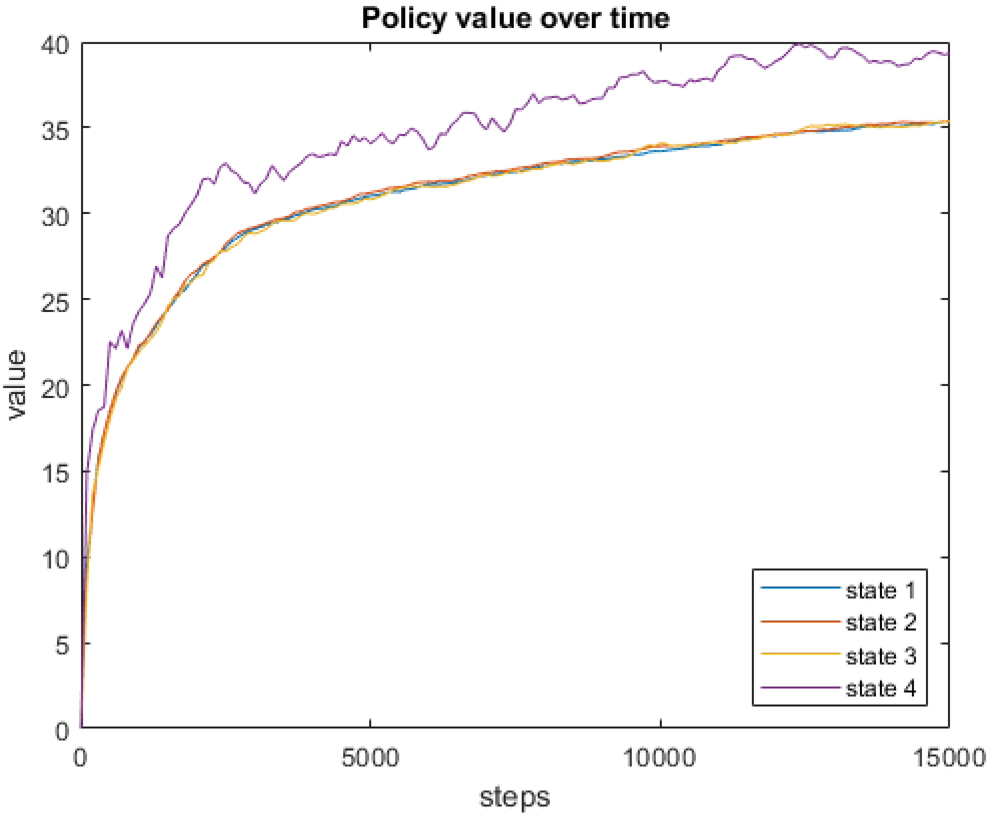


I went for pretty heavy overkill with 200000 steps, and got policy values of $[32.1, 32.5, 32.8, 33.6]^T$.



d.

I tried a policy of $[3, 1, 1, 1]^T$, and after 15000 steps the calculated policy values were $[35.3, 35.4, 35.4, 39.4]^T$.



Since these policy values are higher than the ones from 1c, we have significant evidence that the policy $[3, 1, 1, 1]^T$ is better than $[1, 1, 3, 2]^T$.

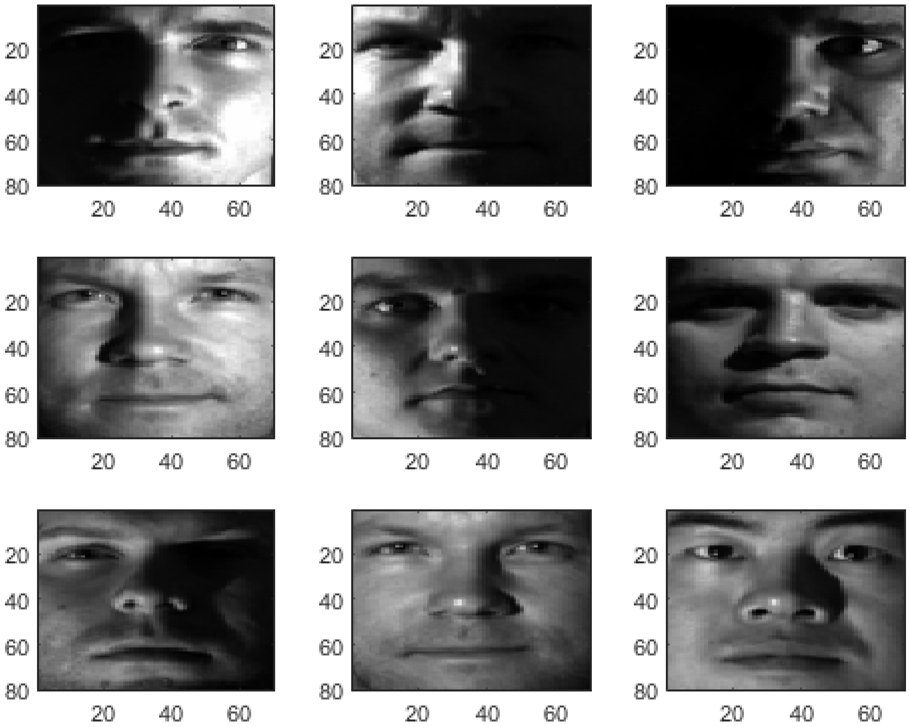
e.
At the end of 15000 steps, my final results for Q were:

		action		
		1	2	3
policy	1	38.72	38.81	39.03
	2	39.22	38.93	38.57
	3	39.03	39.20	40.21
	4	42.31	40.24	38.63

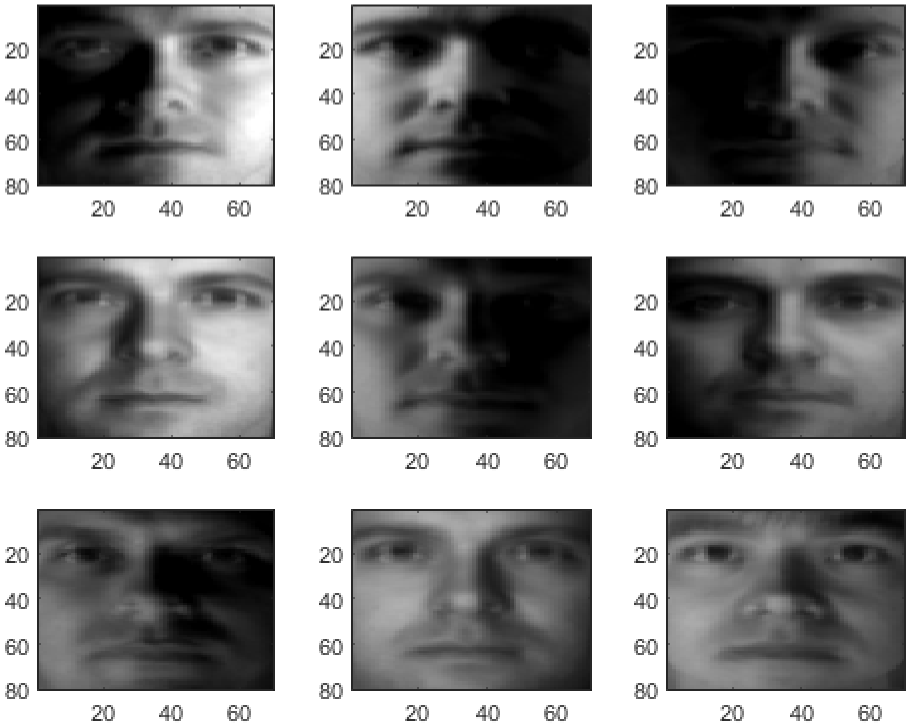
Finding the maximum for each row, we get the policy of $[3, 1, 3, 1]^T$ with policy values of $[39.03, 39.22, 40.21, 42.31]^T$. These policy values are the highest we've yet observed, indicating that this is the best policy we've found.

2 Task 1: Eigenfaces

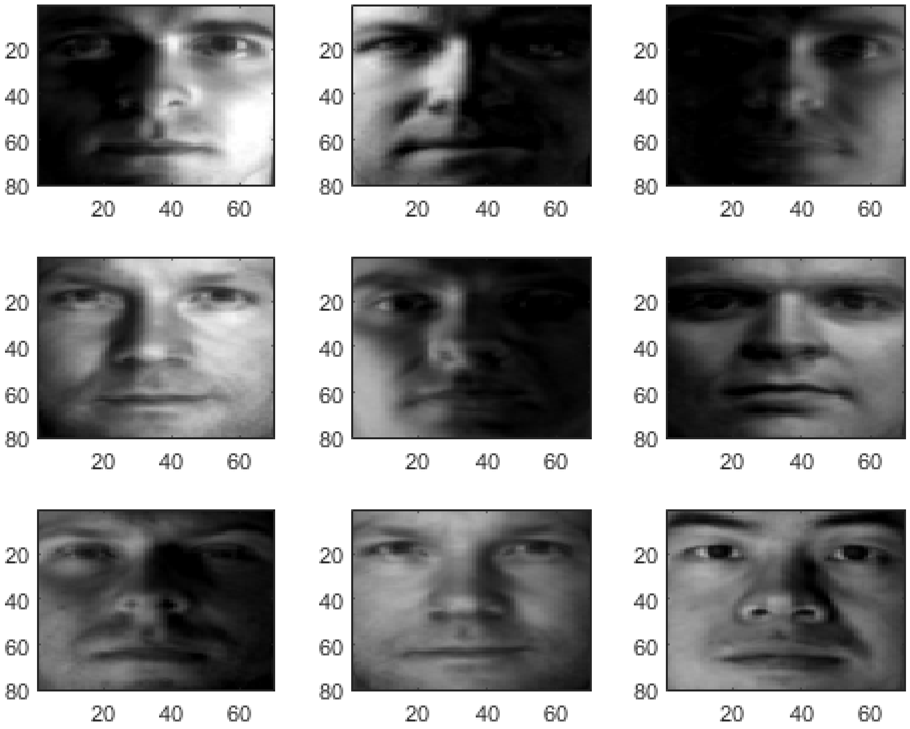
a.
S = [440,115,355,124,295,171,19,103,63]



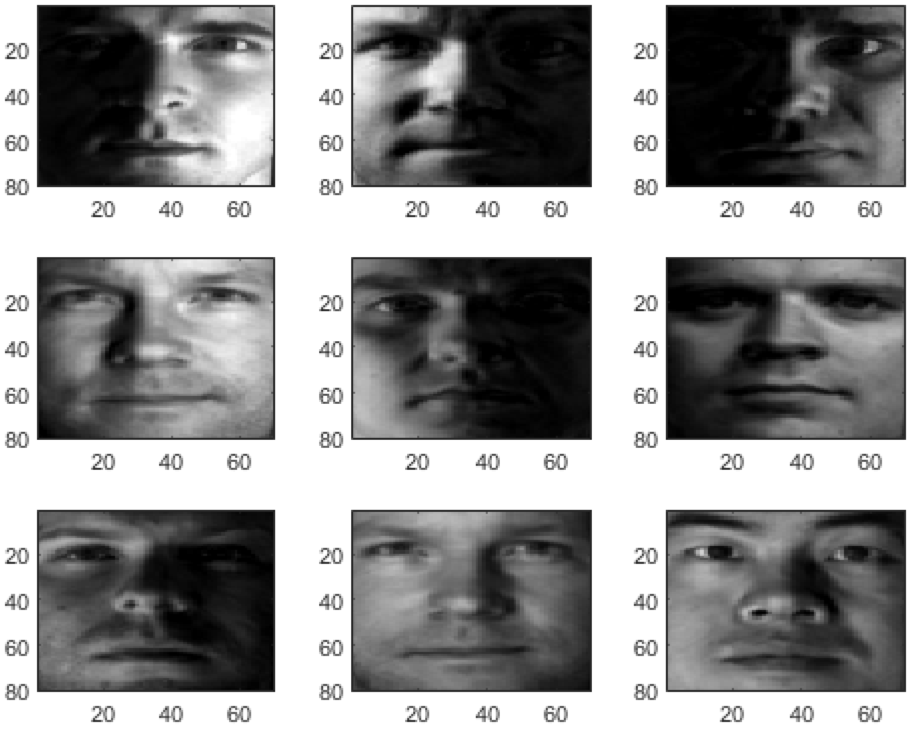
Original Images



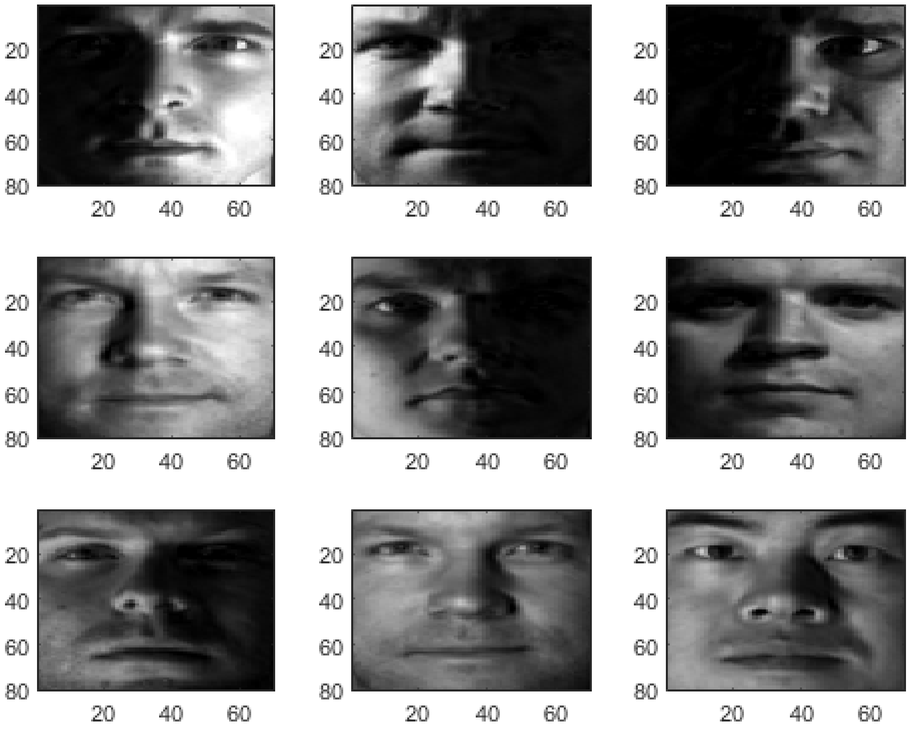
m=5



m=20



m=50

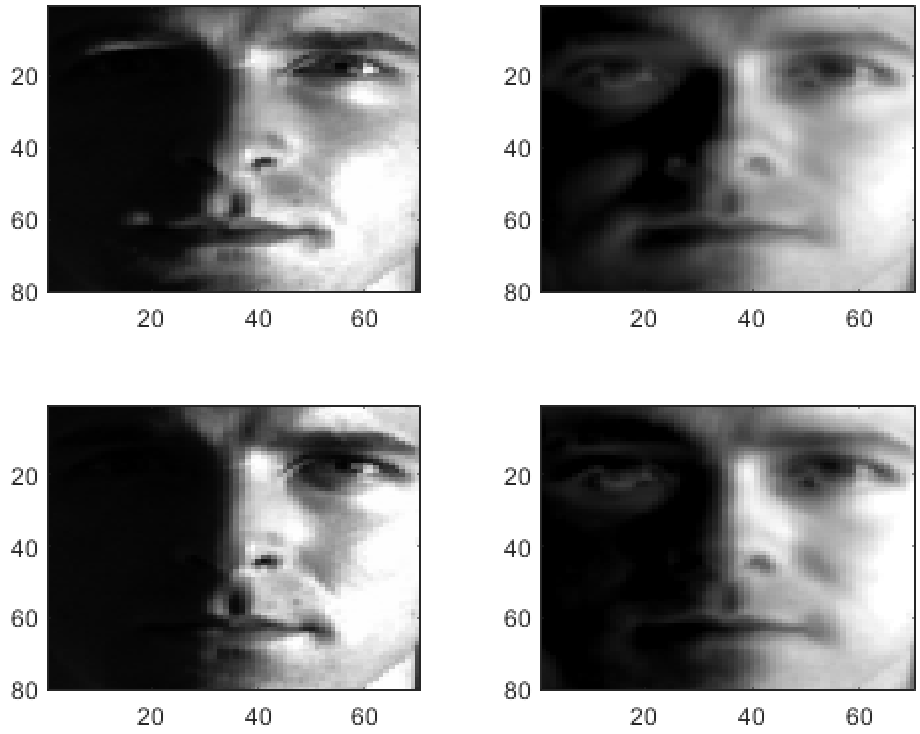


m=100

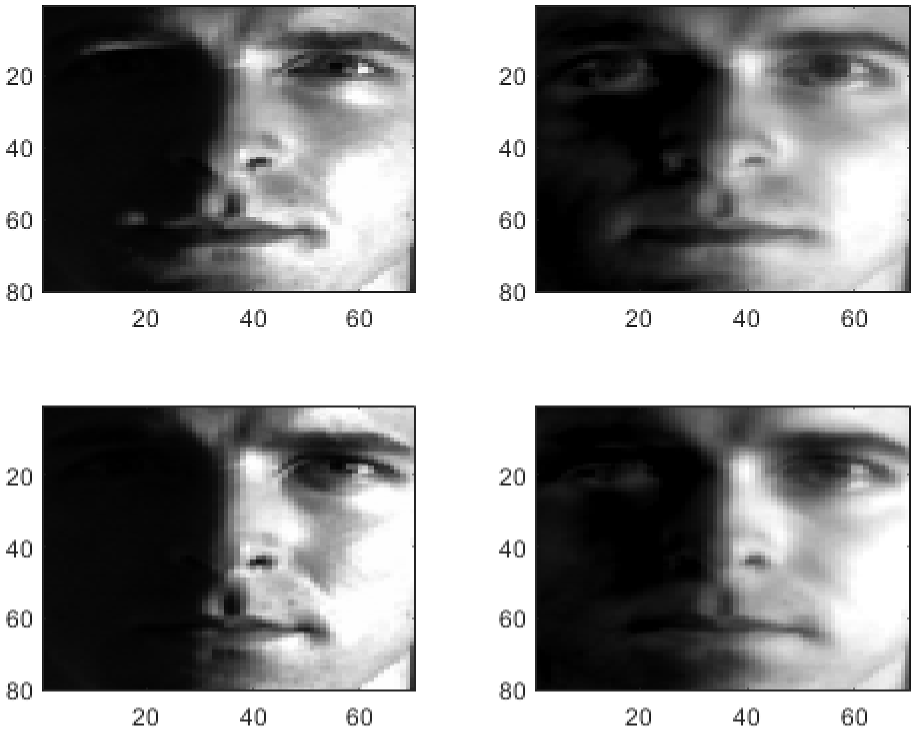
Honestly every reconstruction above m=5 seems to be pretty decent. There are significant changes to m=5 especially around the mouth. While there is an increase in detail between m=20 and m=50, all the minute facial expressions seem to be intact. I'm unable to notice any significant difference between m=50 and m=100.

3 Image 440

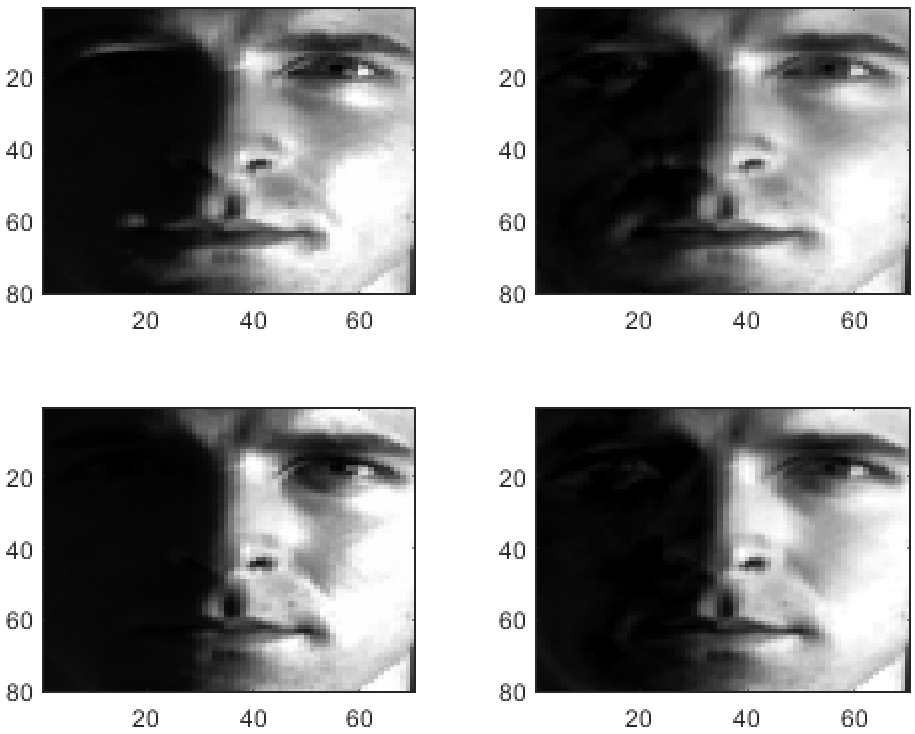
Original Image	Original Reconstructed
Best Match	Best Match Reconstructed



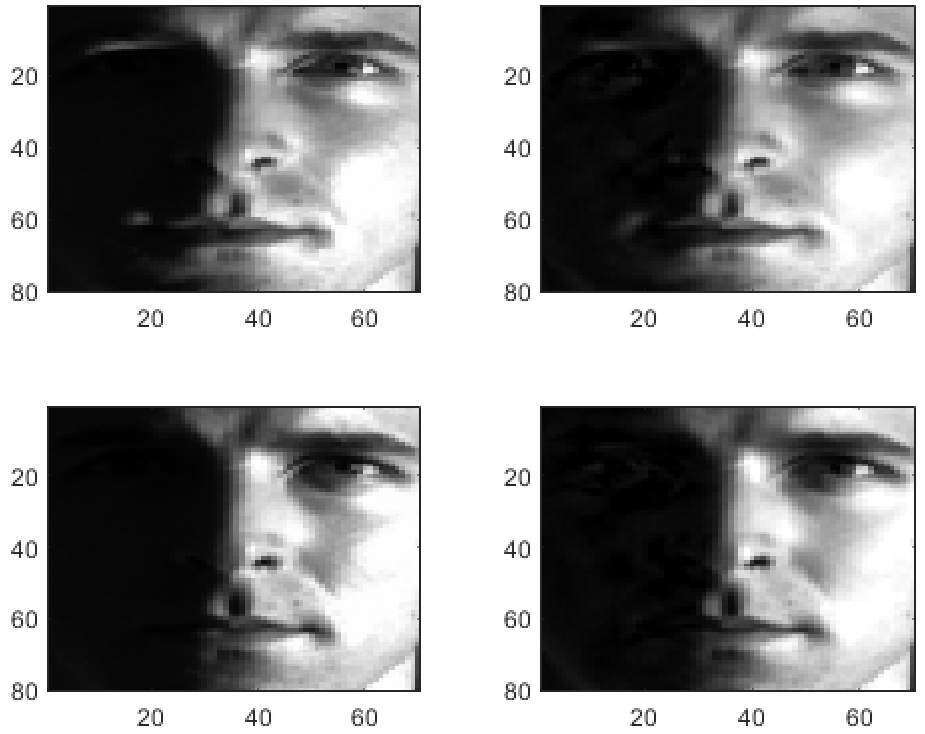
m=5, 440→441



m=20, 440→441

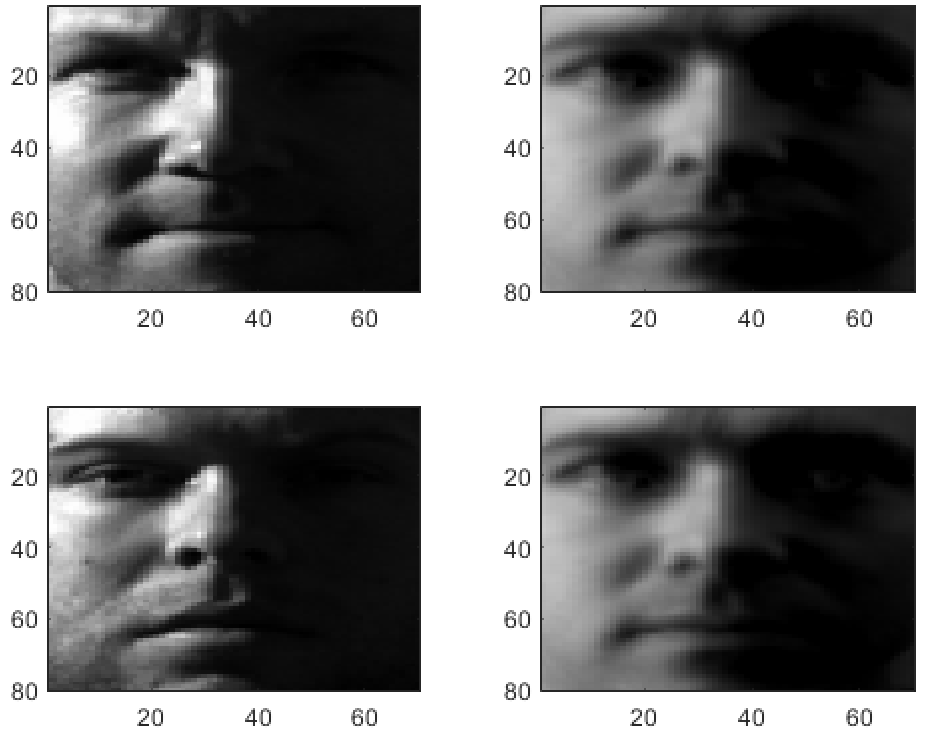


m=50, 440→441

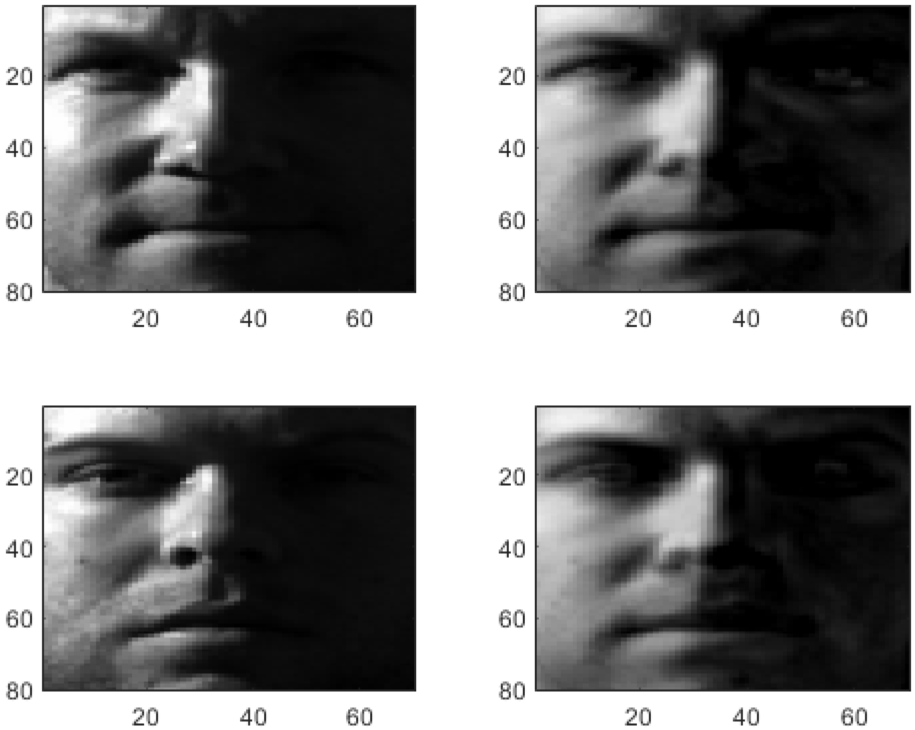


m=100, 440→441

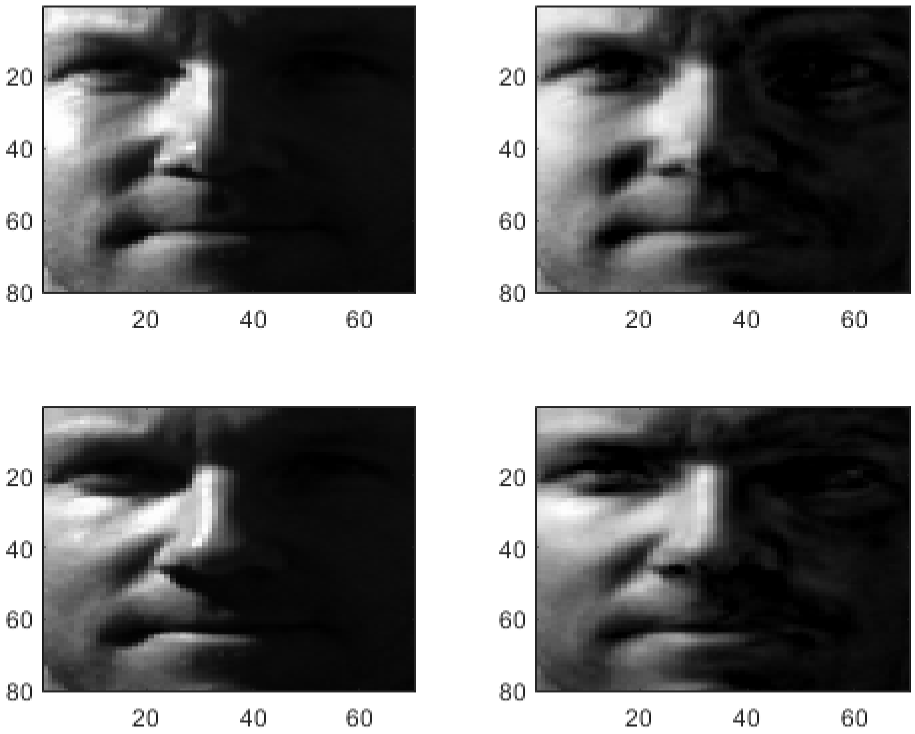
4 Image 115



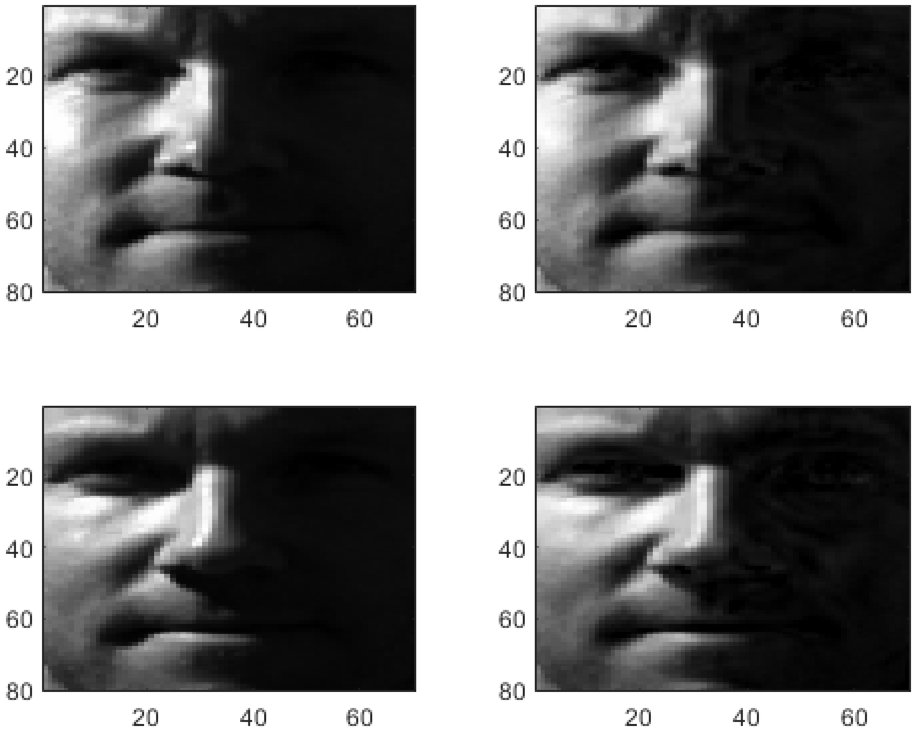
m=5, 115→25



m=20, 115→25

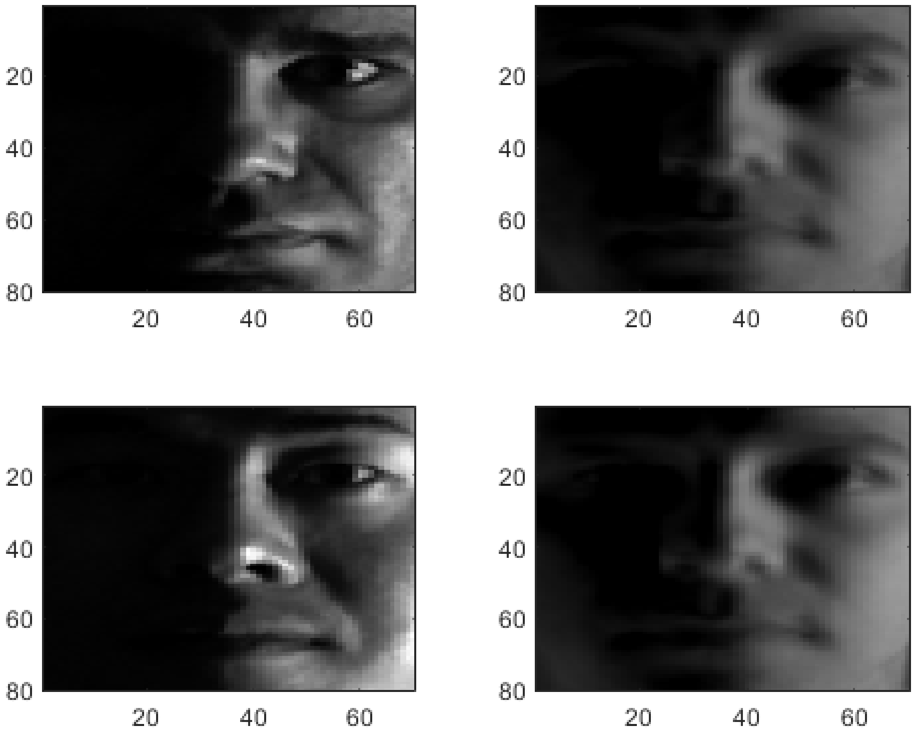


m=50, 115→113

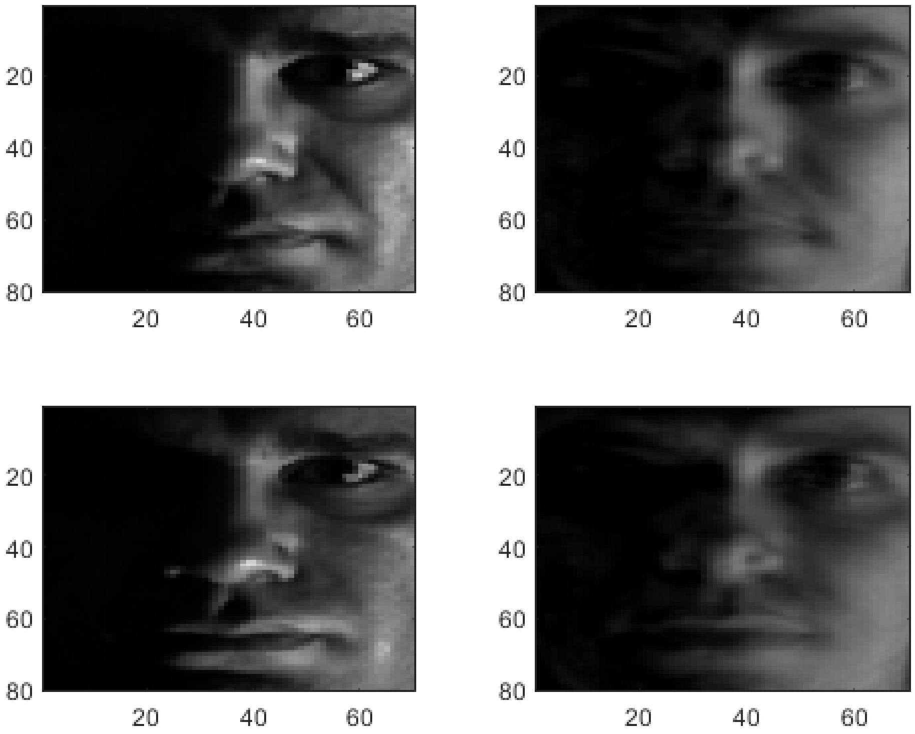


m=100, 115→113

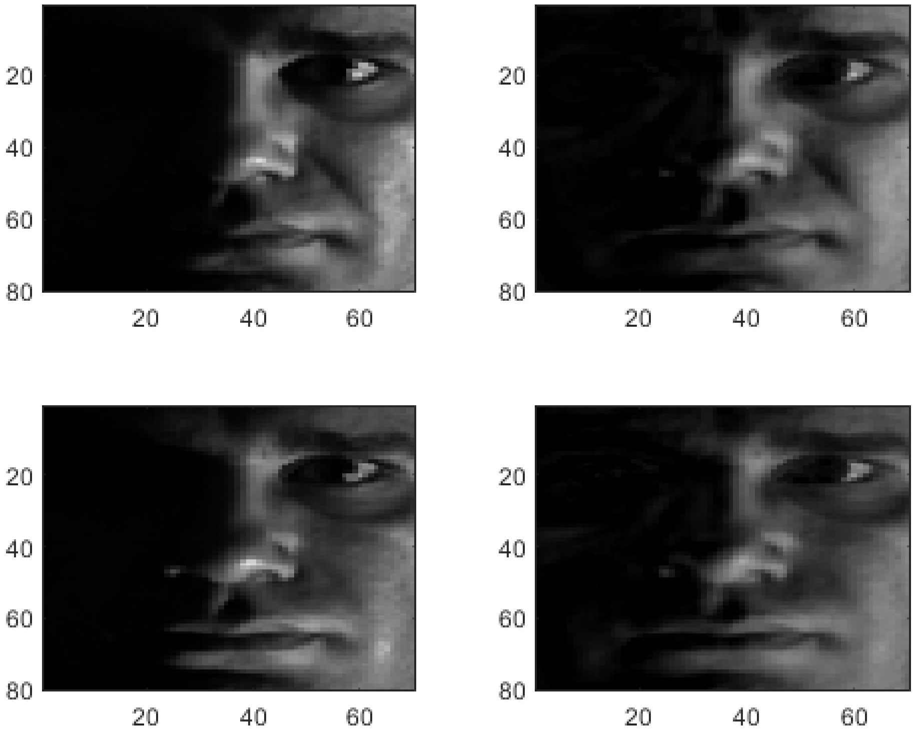
5 Image 355



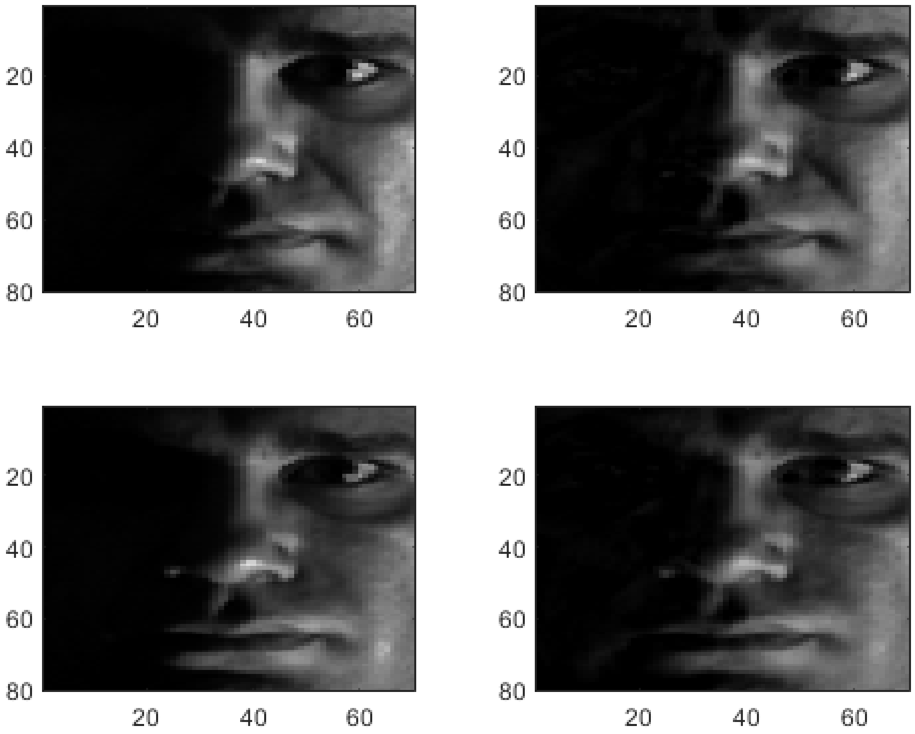
m=5, 355→89



m=20, 355→353

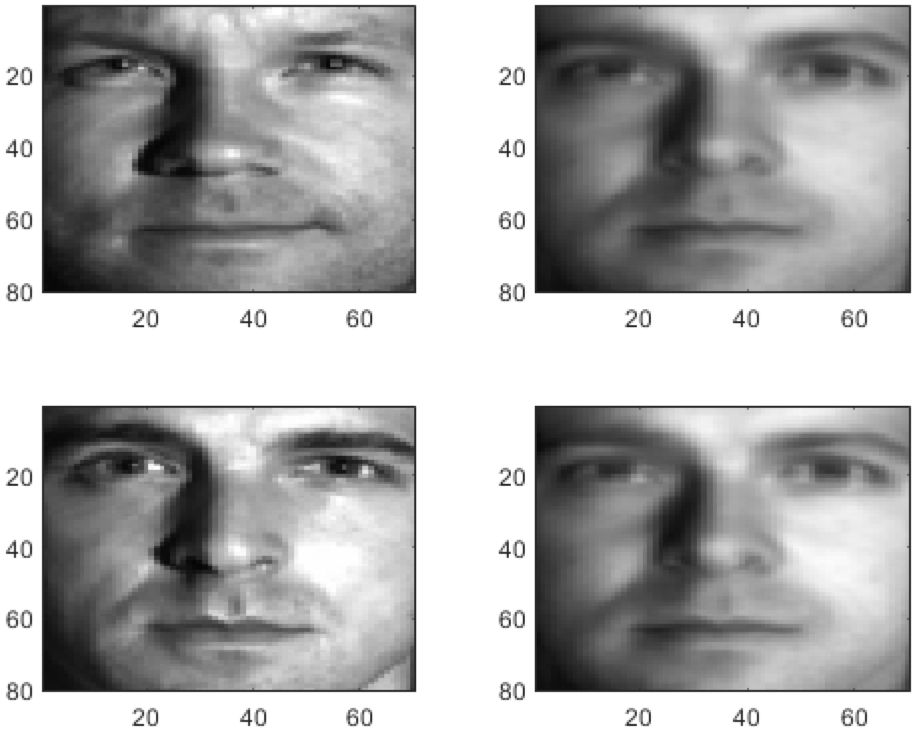


m=50, 355→353

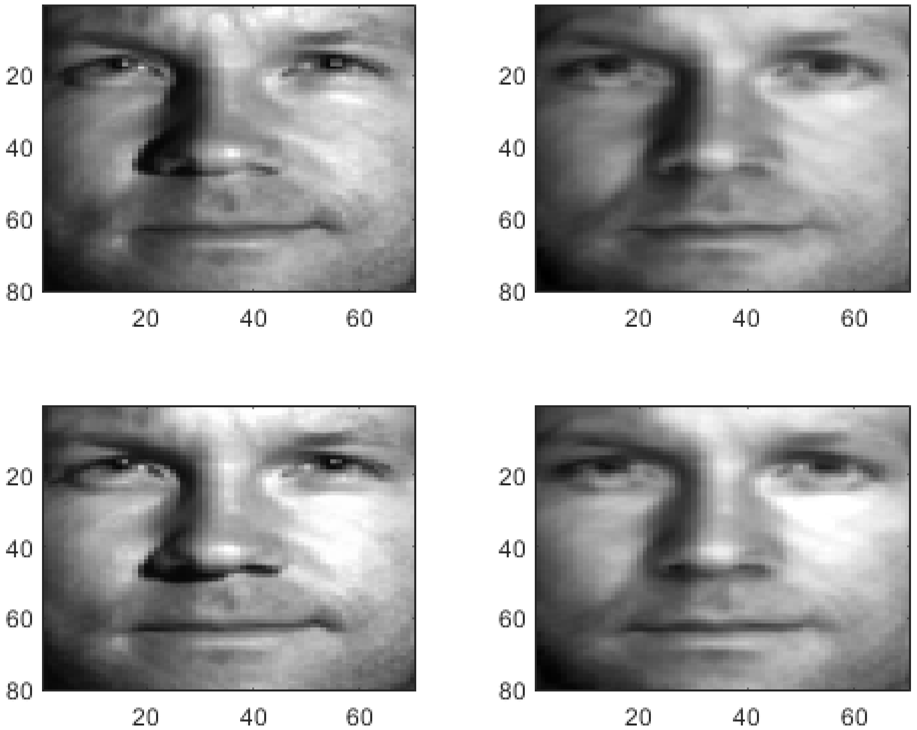


m=100, 355→353

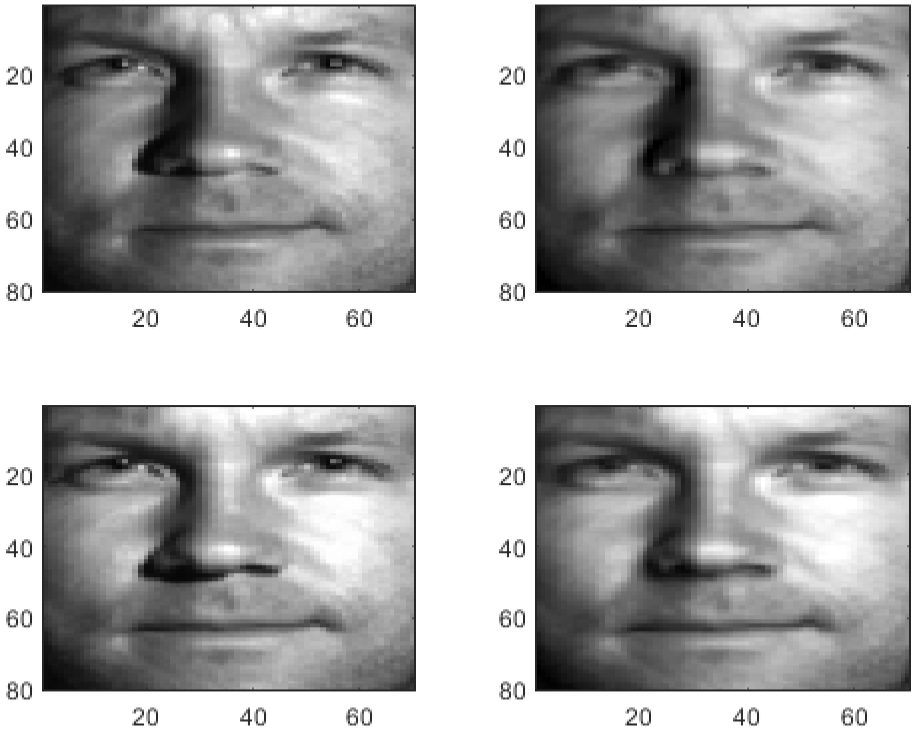
6 Image 124



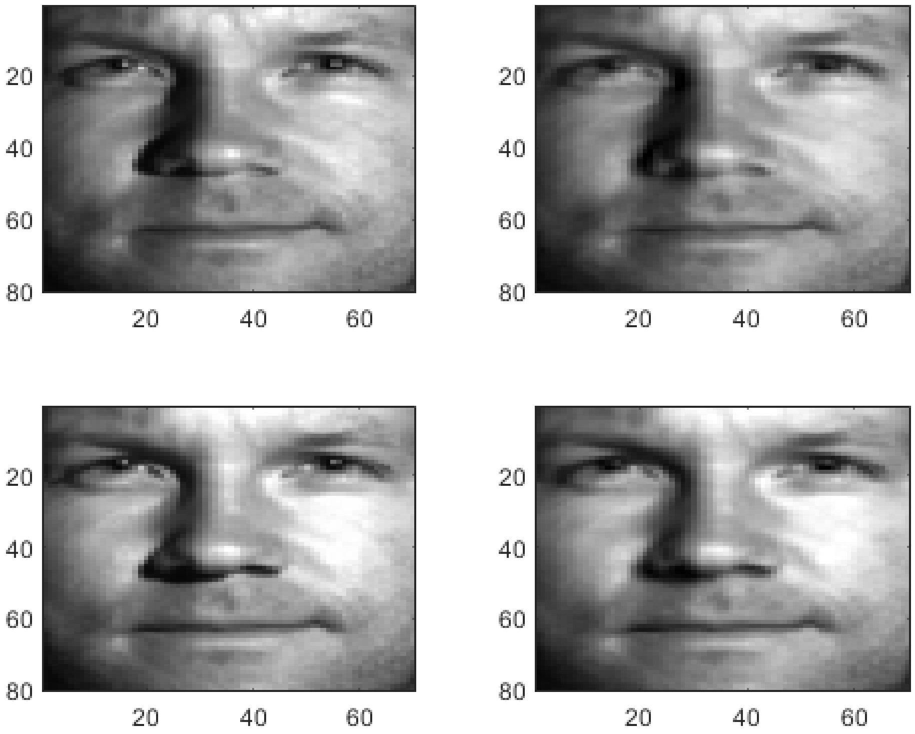
m=5, 124→433



m=20, 124→123

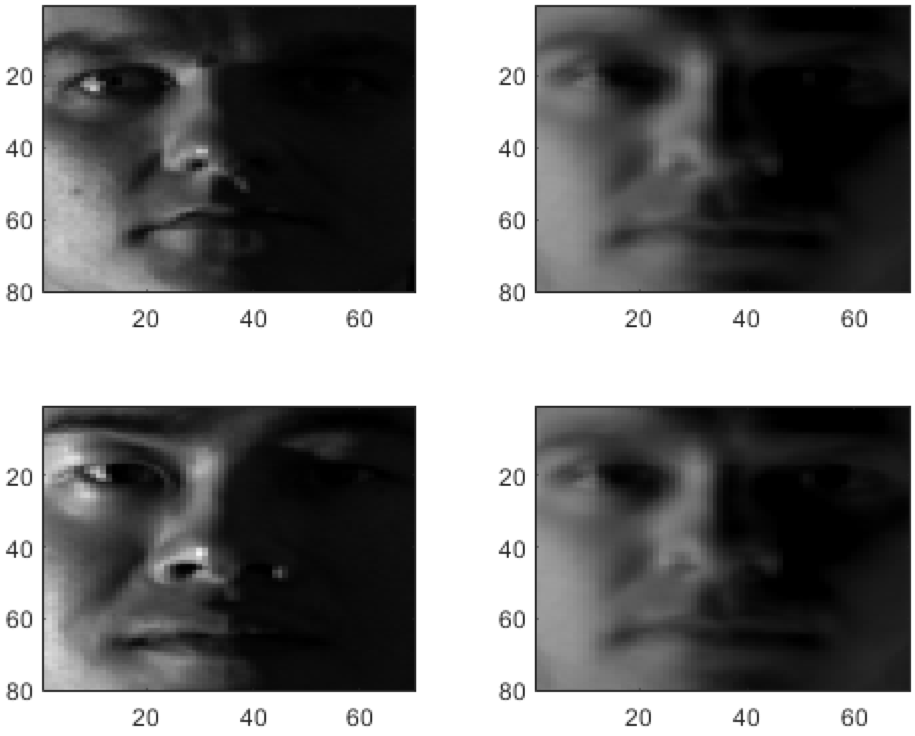


m=50, 124→123

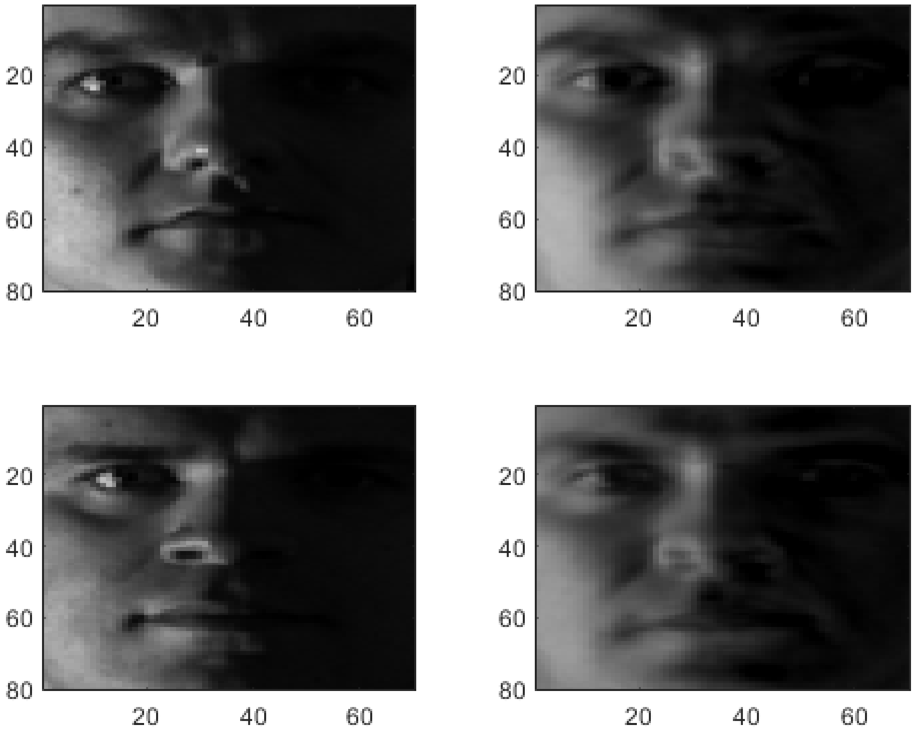


m=100, 124→123

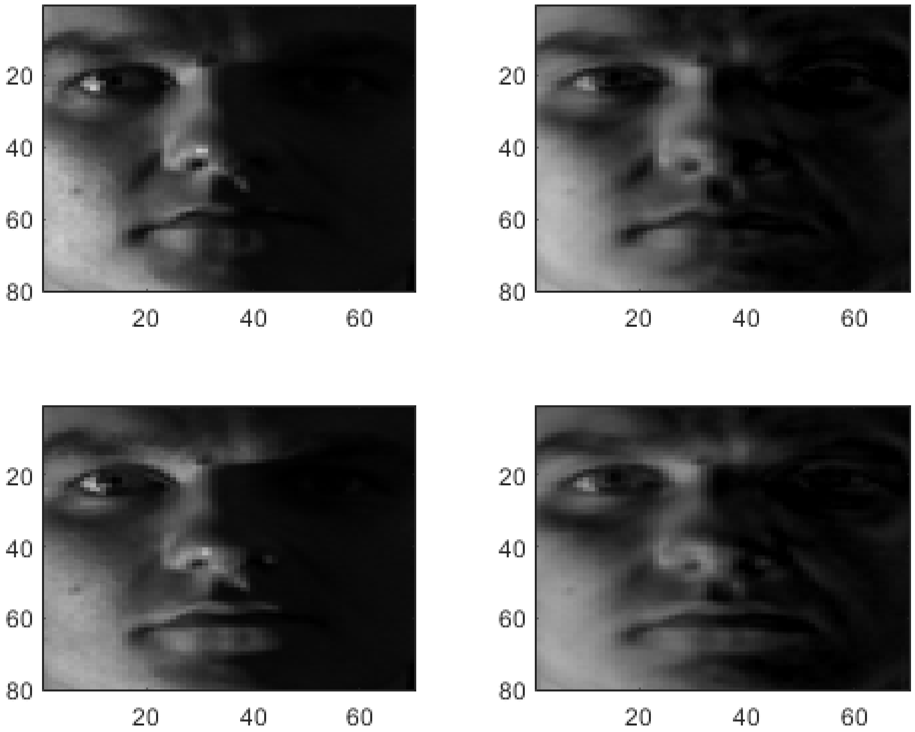
7 Image 295



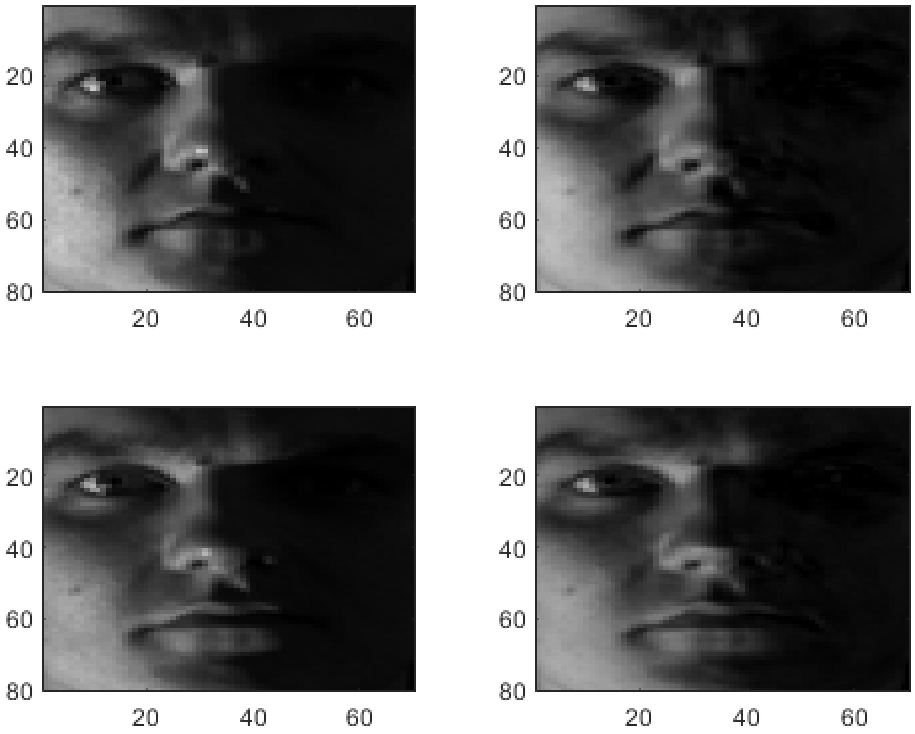
m=5, 295→72



m=20, 295→161

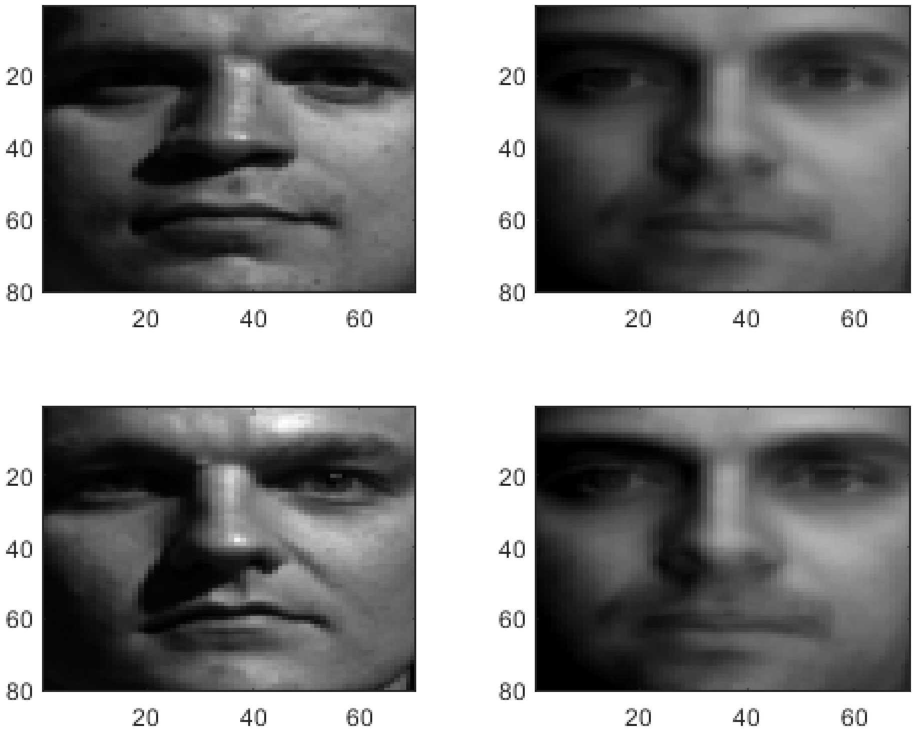


m=50, 295→293

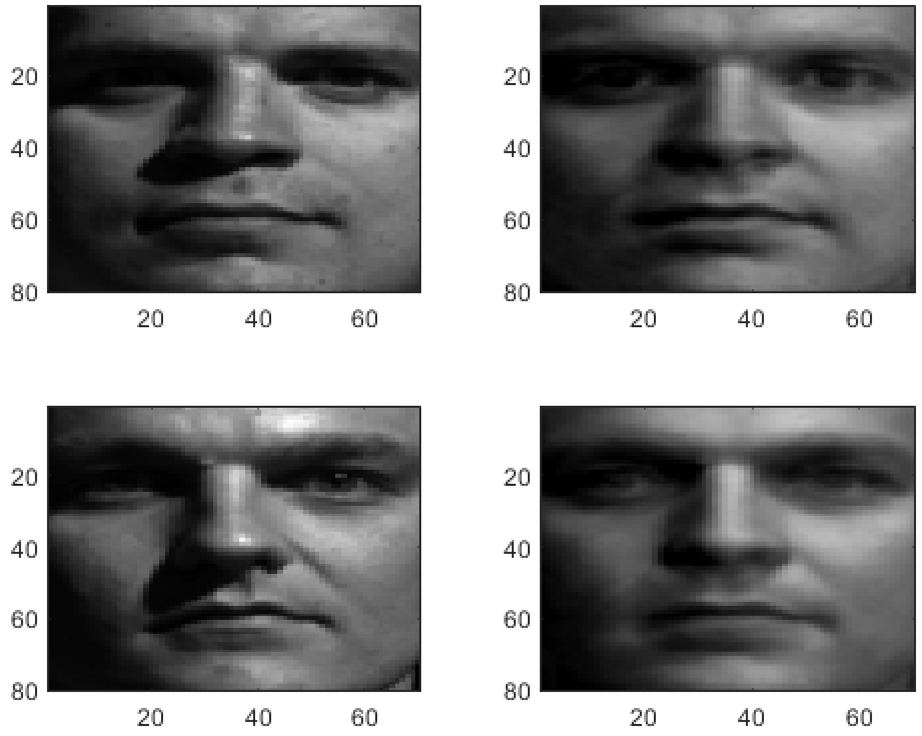


m=100, 295→293

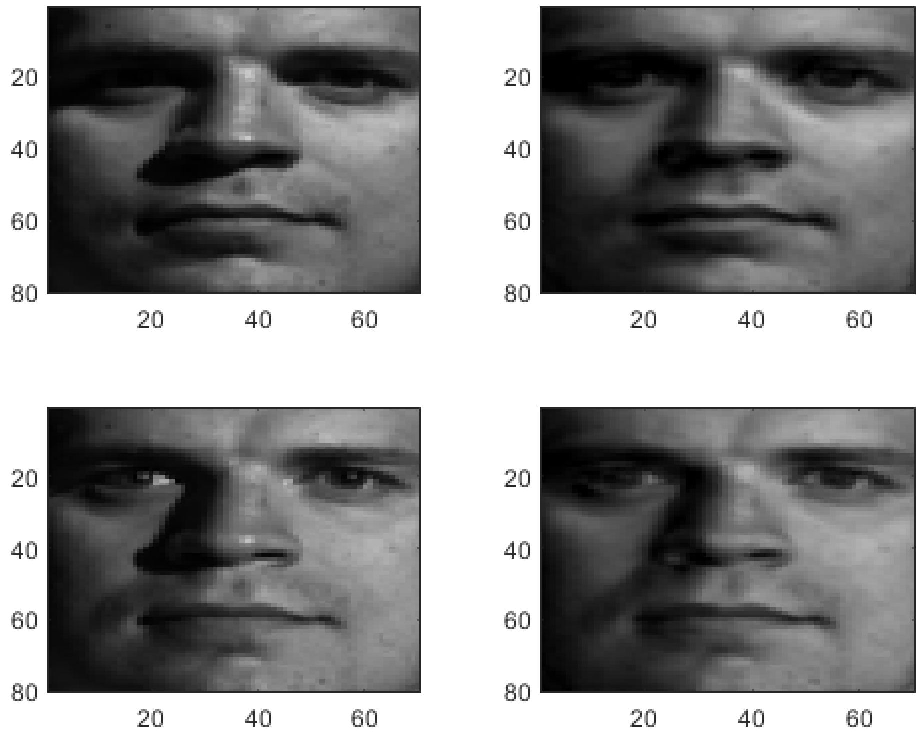
8 Image 171



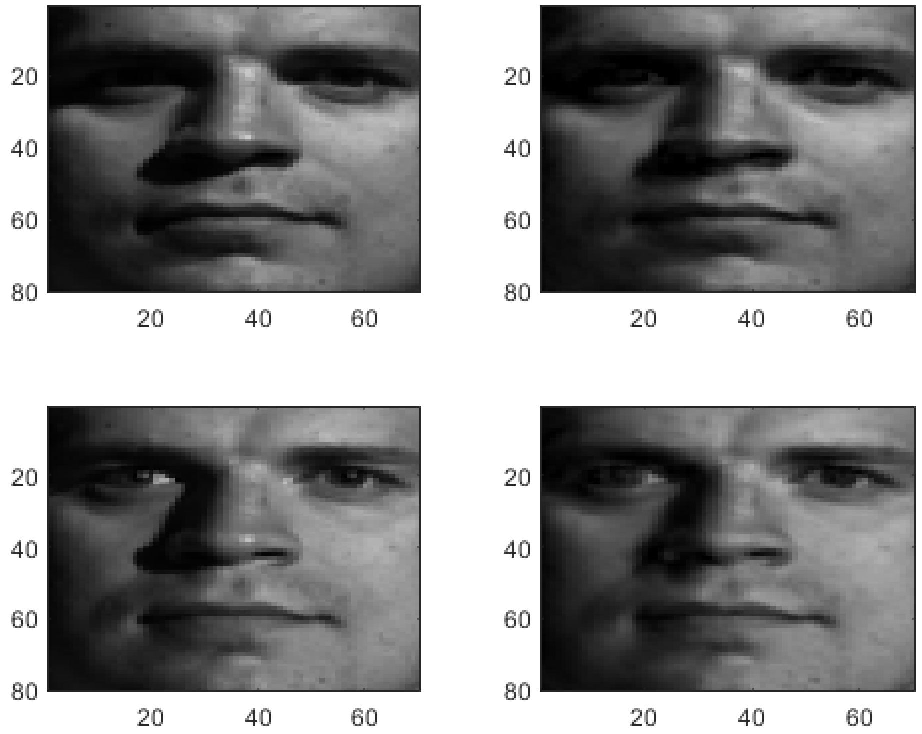
m=5, 171→303



m=20, 171→303

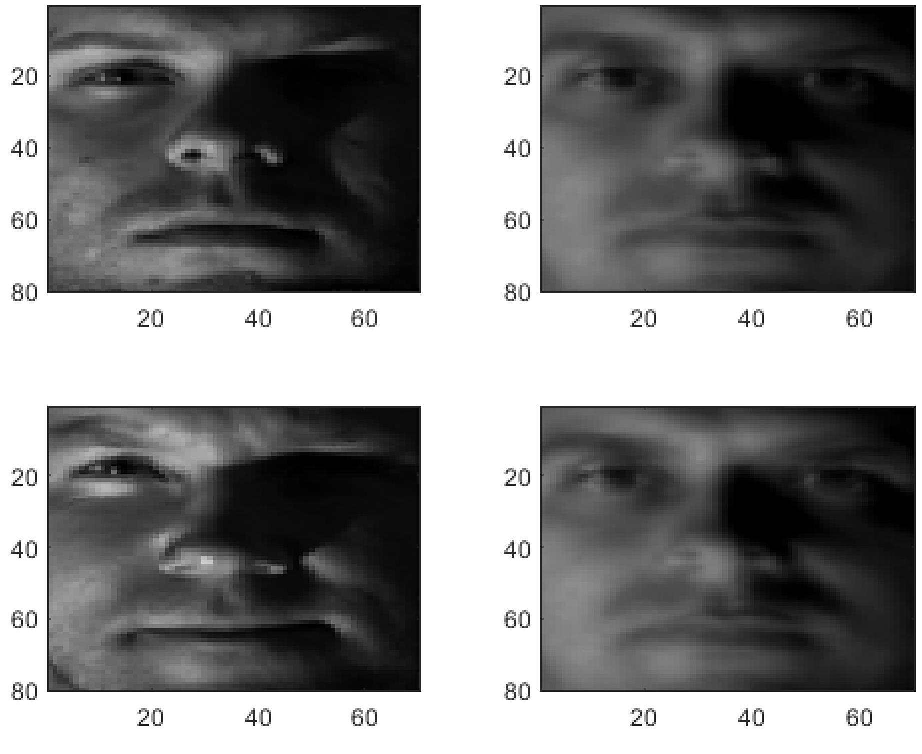


m=50, 171→170

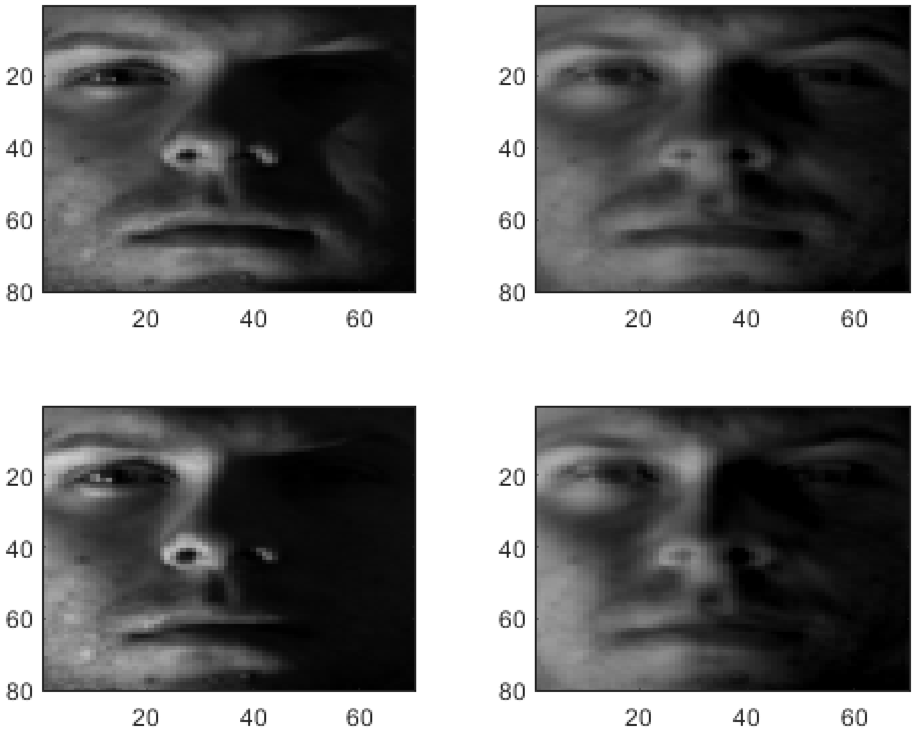


m=100, 171→170

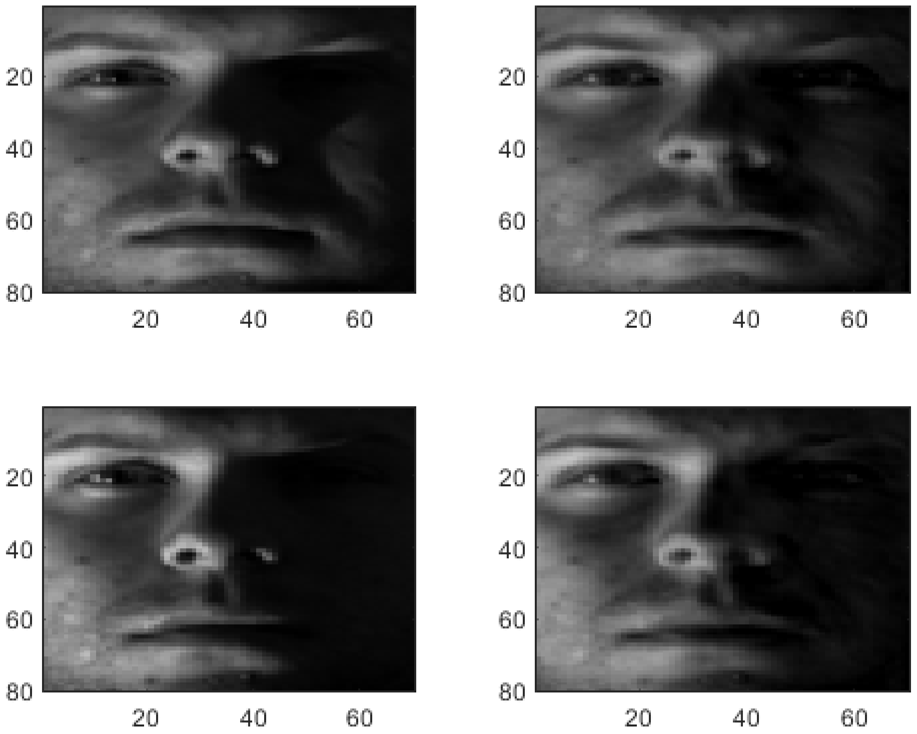
9 Image 19



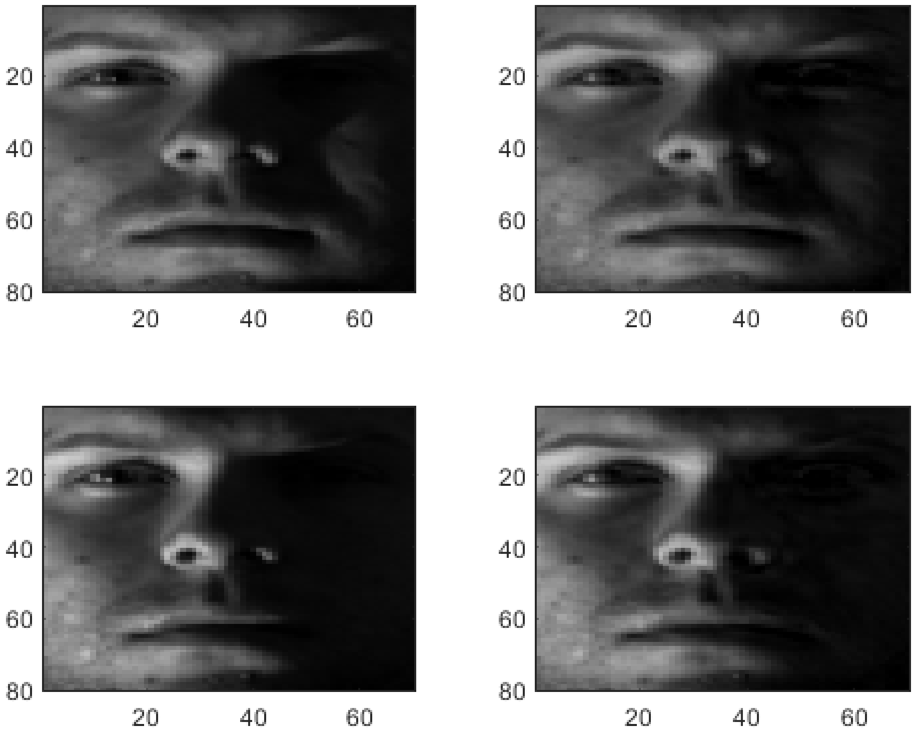
m=5, 19→109



m=20, 19→24

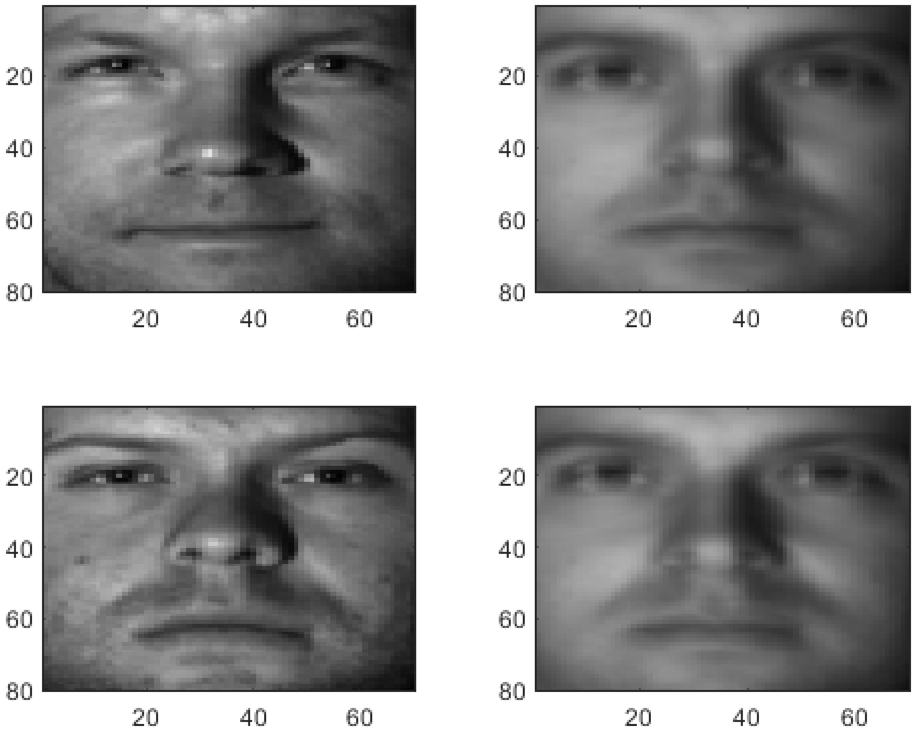


m=50, 19→24

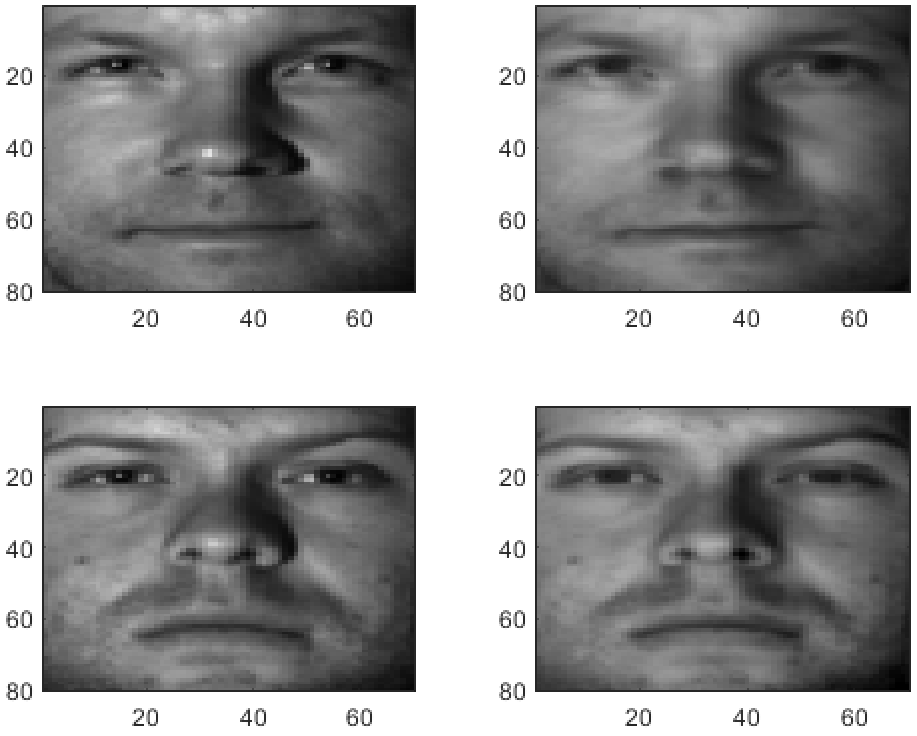


m=100, 19→24

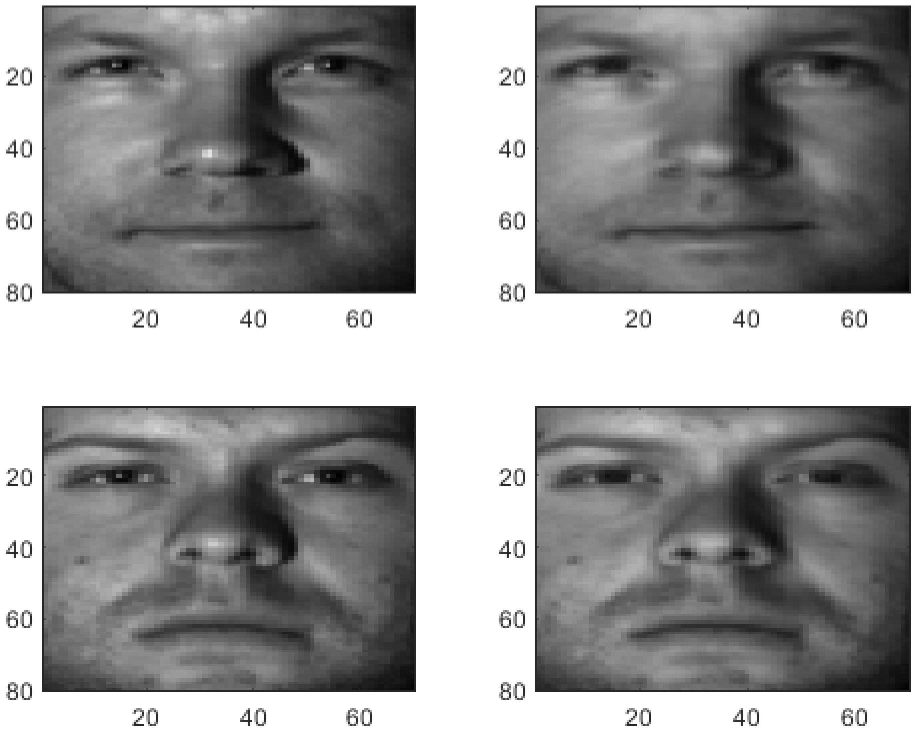
10 Image 103



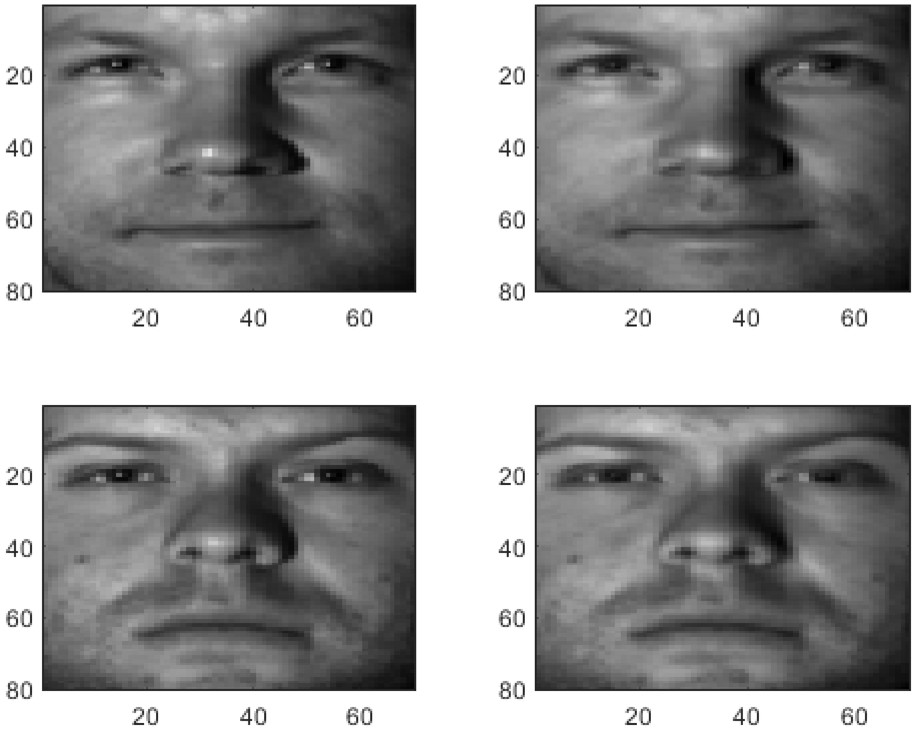
m=5, 103→12



m=20, 103→12

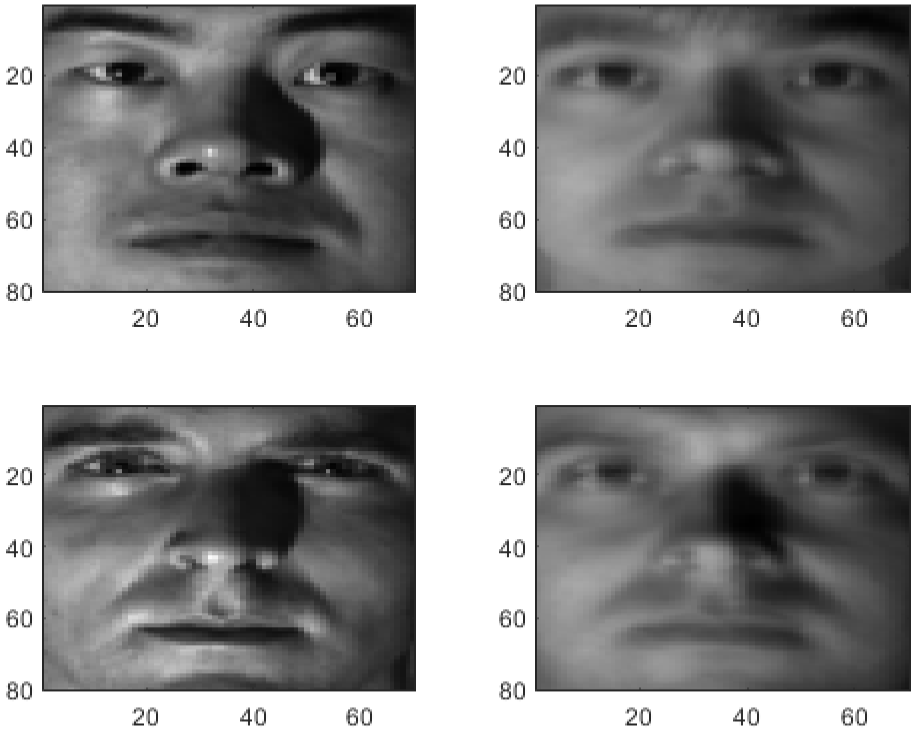


m=50, 103→12

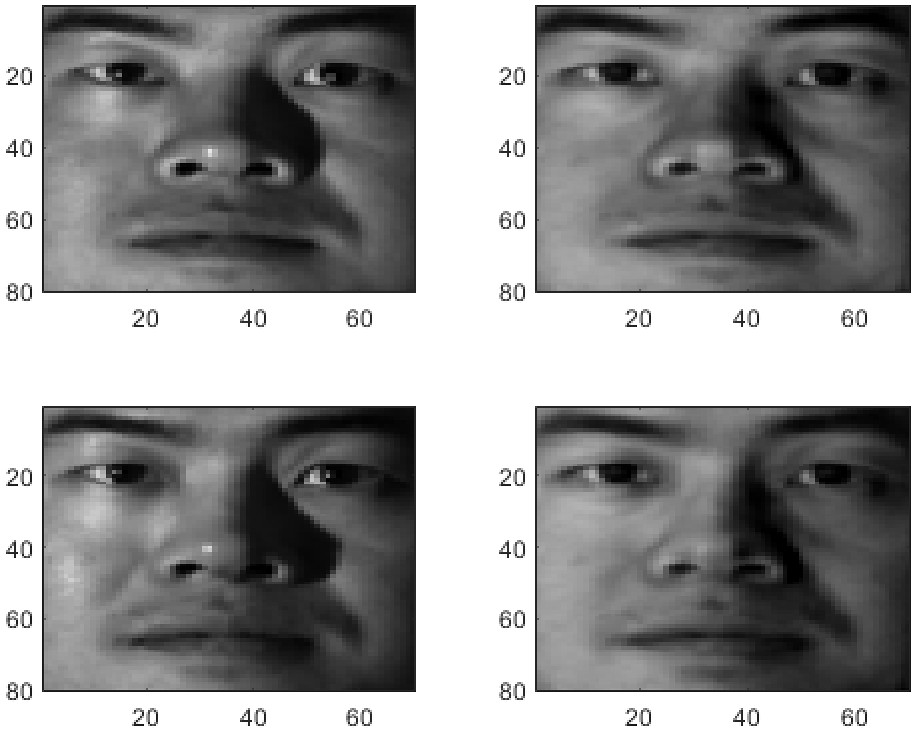


m=100, 103→12

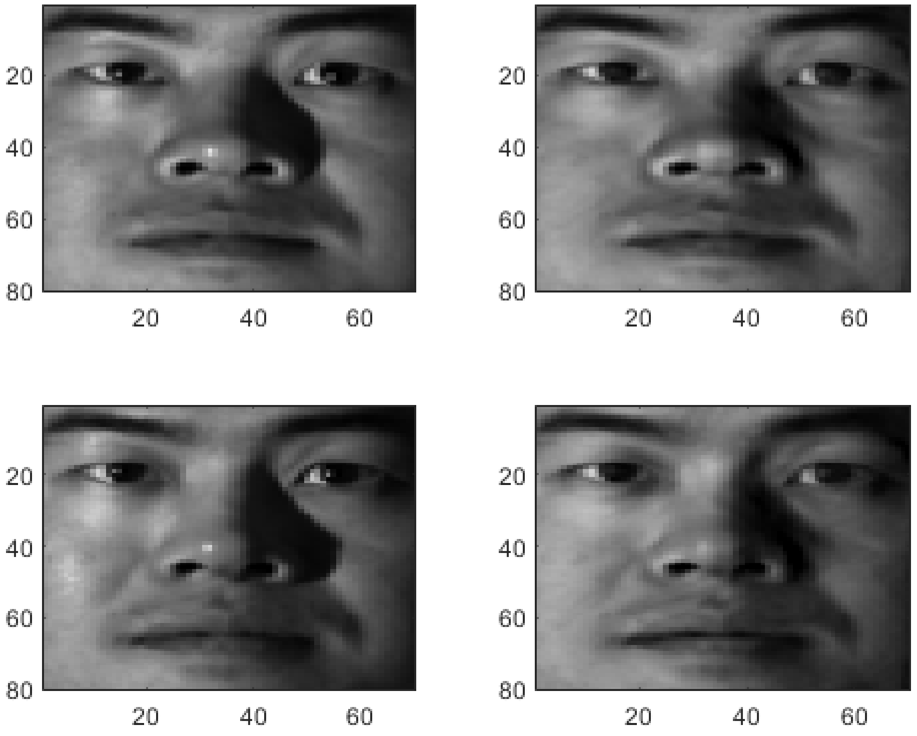
11 Image 63



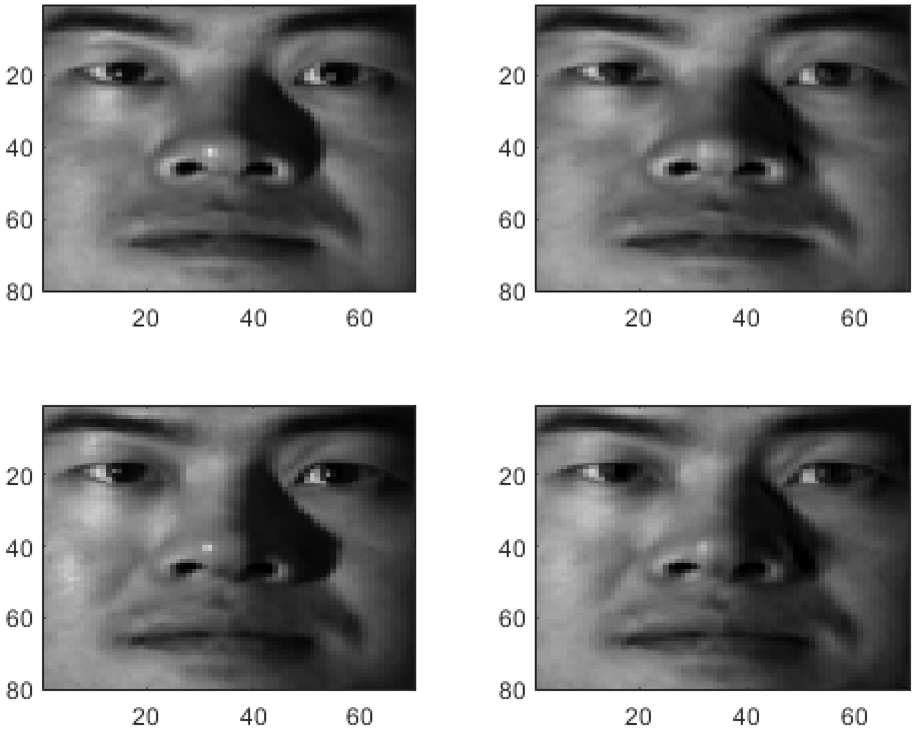
m=5, 63→412



m=20, 63→64



m=50, 63→64



m=100, 63→64

12 Eigenfaces: Task 2 Analysis

- image 440
  - matched with image 441 for all m
- image 115
  - matched with image 25 for m = 5, 20
  - matched with image 113 for m = 50, 100
- image 355
  - matched with image 89 for m = 5
  - matched with image 353 for m = 20, 50, 100
- image 124
  - matched with image 433 for m = 5
  - matched with image 123 for m = 20, 50, 100
- image 295
  - matched with image 71 for m = 5
  - matched with image 161 for m = 20
  - matched with image 293 for m = 50, 100
- image 171
  - matched with image 303 for m = 5, 20
  - matched with image 170 for m = 50, 100

- image 19
  - matched with image 109 for m = 5
  - matched with image 24 for m = 20, 50, 100
- image 103
  - matched with image 12 for all m
- image 63
  - matched with image 412 for m = 5
  - matched with image 64 for m = 20, 50, 100

Of the 9 images I looked at, the only one that did not eventually match with the same subject was image 103, which matched with image 12 for all values of m. At m=5, 1/9 images matched with the correct subject. At m=20, 5/9 images matched with the correct subject. At m=50 and m=100, 8/9 images matched with the correct subject. At m=20, 5/9 images matched with the correct subject.

With the exception of image 103, all other images matched up with the correct subject at m=50, while just over half did so at m=20. The matching obviously improves as m increases, but sees diminishing returns (if any) after m=50.

13 Task 3: knn

In order to understand the nearest-neighbors algorithm, I imagine all the elements in the training data plotted in n-dimensional (or in the case of this problem, m-dimensional space). Then, to determine the class of a new element, calculate the euclidean distances from it to each training element and predict it will have the same class as the element that's the smallest Euclidean distance away. Essentially, the new element will be predicted to have *the same class as the closest known element.*

However, the nearest-neighbor classifier has a significant weakness: it is extremely vulnerable to outliers in the training data skewing the results. For example, imagine the training data has two classes A and B. All the elements of A are tightly packed within 1 unit of each other. 5 units away, all units of B are packed in a similarly tight configuration, except for a single element of B that exists in the middle of the A cluster. Under the nearest-neighbor classifier, a test element adjacent to this outlier would be labelled as class B, despite having far more

To fix this, instead of just looking at the closest neighbor, K nearest-neighbor fixes the outlier problem by expanding the range of elements to consider until K elements of the same class are found. Now, the predicted class won't be swayed by 1 or 2 (or any number less than k) outliers.

14 Task 4: Applied knn

I ran the knn classifier for k values of 1, 5, 10, 20, and 40, with m-values of 5, 20, 50, and 100. Note that for k=1, the training data will have no error since the image will be closest to itself.

For the training data, I got misclassification errors of:

	k=1	2	5	10	20	40
m=5	-	0.42	0.52	0.53	0.59	0.58
20	-	0.17	0.12	0.17	0.22	0.34
50	-	0.08	0.04	0.08	0.16	0.26
100	-	0.06	0.04	0.07	0.16	0.26

For the testing data, I got misclassification errors of:

	k=1	2	5	10	20	40
m=5	0.82	0.78	0.74	0.68	0.66	0.7
20	0.34	0.36	0.3	0.38	0.38	0.42
50	0.16	0.24	0.26	0.26	0.32	0.4
100	0.14	0.22	0.22	0.28	0.34	0.4

It's difficult to determine the exact optimal values for m and k. One surprise is that increasing m and (especially) k doesn't always decrease the testing error after m=20 and k=10. Oddly enough, the lowest testing error I found was for m=100 and k=1. As I explained in the previous section, k>1 will decrease the effect of solitary outliers, but for this particular training and testing set that doesn't seem to be an issue. Noticing how well k=1 performed actually inspired me to go back and test k=2 to add some granularity.

Even though m=5 k=1 was the highest performing for this specific dataset, if we were to continue I might try m=50 or 100 and k=2 or 5.