

Eigenfaces and Face Recognition

전기공학과
2016027638 함재형

Contents

- Basic Theory Review
- Introducing the assignment
- Works
- Results
- Analysis
- Conclusions

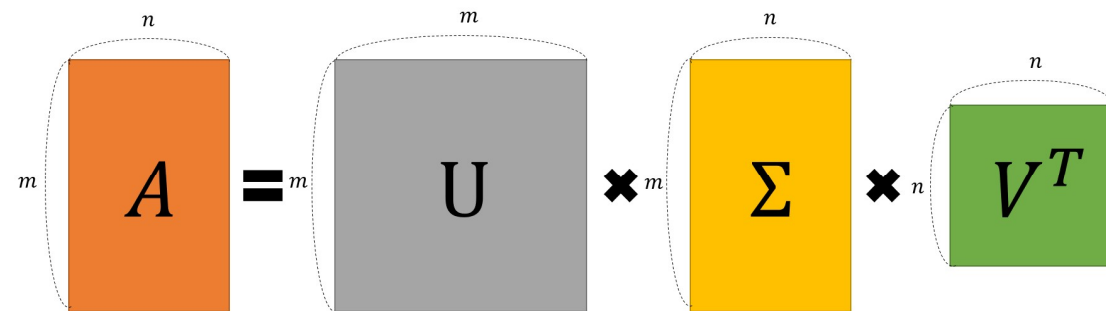
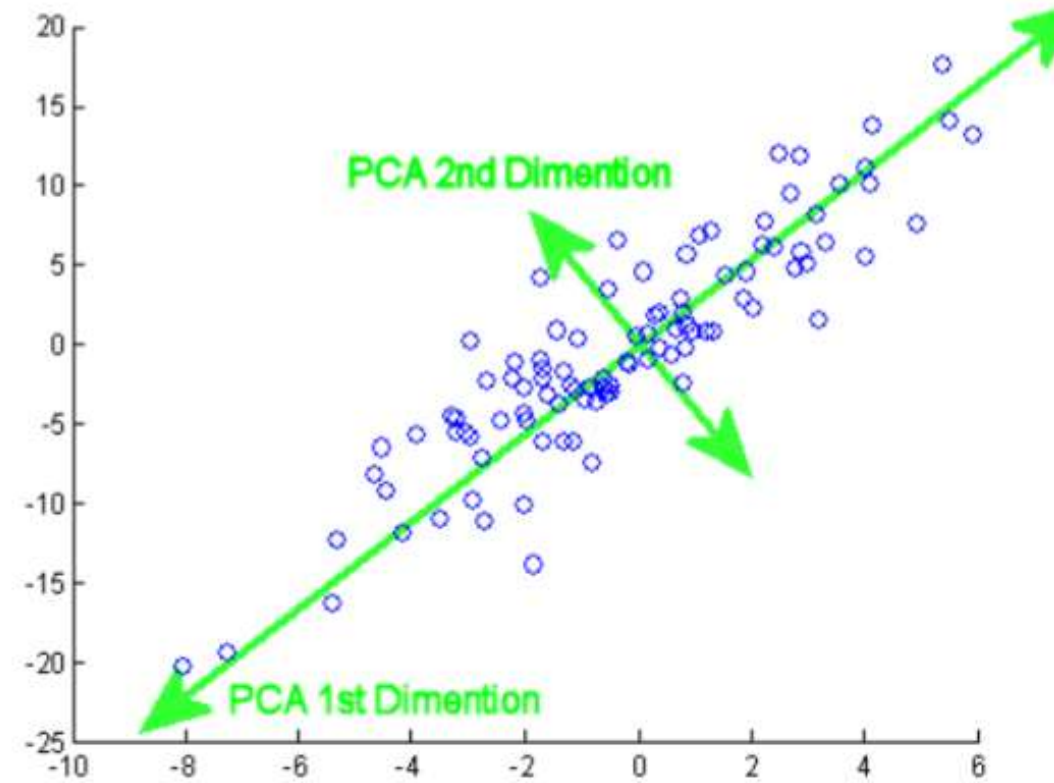
Basic Theory Review

- Eigenface: 얼굴 사진에서의 basis 벡터 이미지
- PCA(Principal Component Analysis)를 이용
- PCA: Covariance 행렬에 SVD를 적용하여 얻은 eigenvalue 중 값이 큰 것을 찾는다



Basic Theory Review

- SVD: $A = U\Sigma V^T$ 로 값을 분해
- $U: [U_1 \ U_2]$
- $U_1: A^T A$ 의 column space의 basis
- $U_2: A^T A$ 의 left null space의 basis
- $\Sigma: A^T A$ 의 특이 값 행렬
- $V^T: [V_1 \ V_2]^T$
- $V_1: A^T A$ 의 row space의 basis
- $V_2: A^T A$ 의 null space의 basis



Introducing the assignment

과제: PCA를 이용하여 Eigenfaces를 찾고
계수 분석 후 Face Recognition을 해본다.

Works – Finding eigenfaces

(1) 12288개의 64x64 grey face image, 4096개의 eigenface

```
41 def pca(X, n_pc):  
42     """Find mean"""  
43     mean = np.mean(X, axis=0)  
44     """Center data"""  
45     centered_data = X - mean  
46     """SVD"""  
47     U, S, V = np.linalg.svd(centered_data)  
48     """Eigenfaces"""  
49     components = V[:n_pc]  
50  
51     return components, mean, centered_data
```

PCA

```
67     """Coefficients"""  
68     weights = np.dot(Y, C.T)
```

Coefficient

```
70 def reconstruction(Y, C, M, h, w, image_index, weights, X, n_components):  
71  
72     centered_vector = np.dot(weights[image_index, :], C)
```

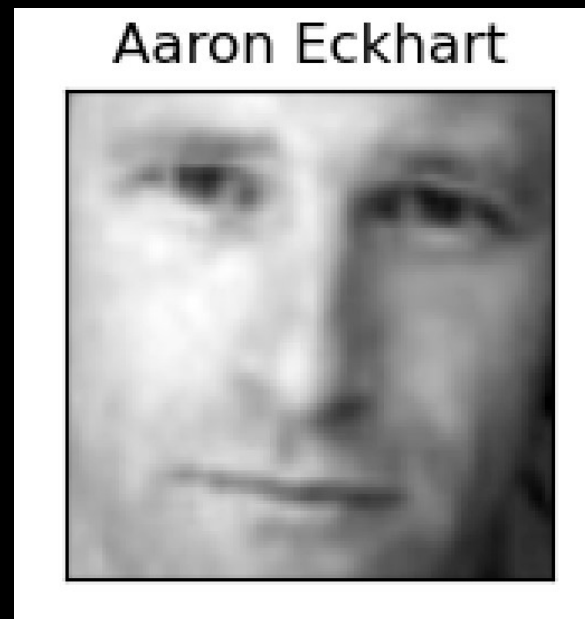
Reonstrcution

Results – Finding eigenfaces

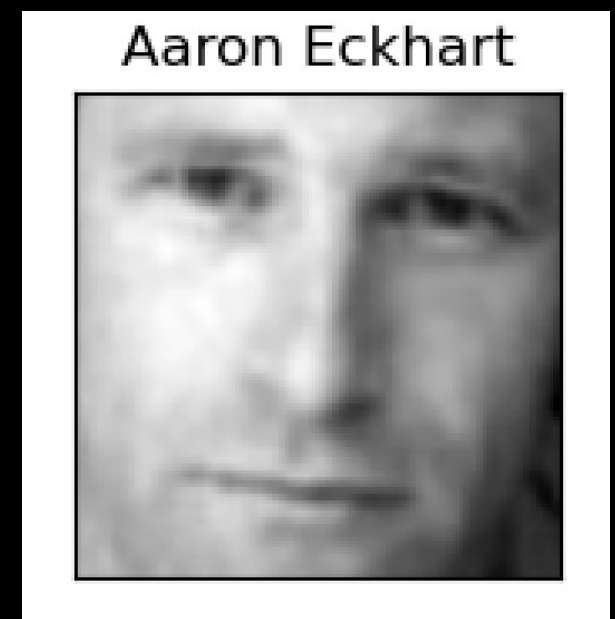
(1) 12288개의 64x64 grey face image, 4096개의 eigenface



4096 different eigenfaces



Original



Reconstructed

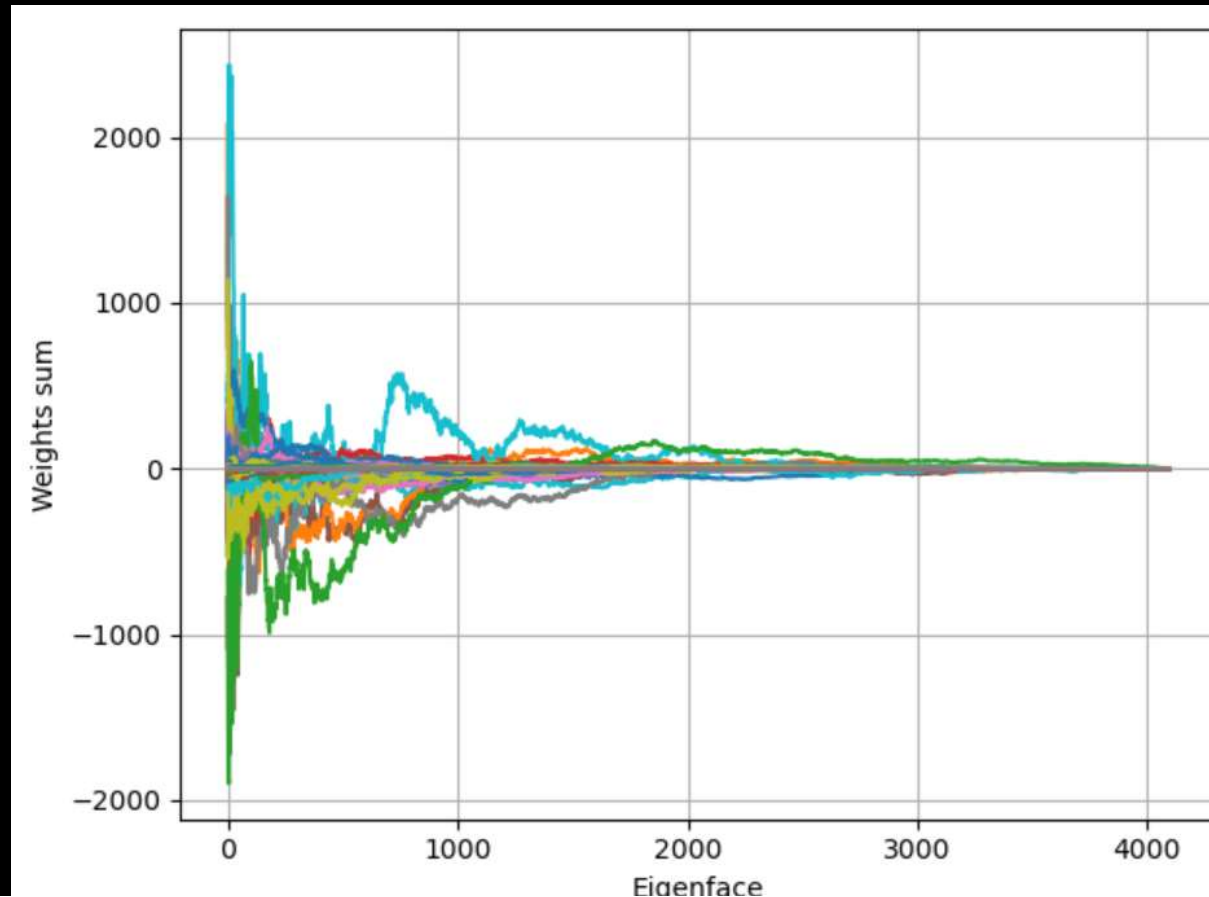
이미지 차원과 동일한 개수의 eigenface ➡ 완벽에 가까운 reconstruction

Analysis – Finding eigenfaces

(1) 12288개의 64x64 grey face image, 4096개의 eigenface

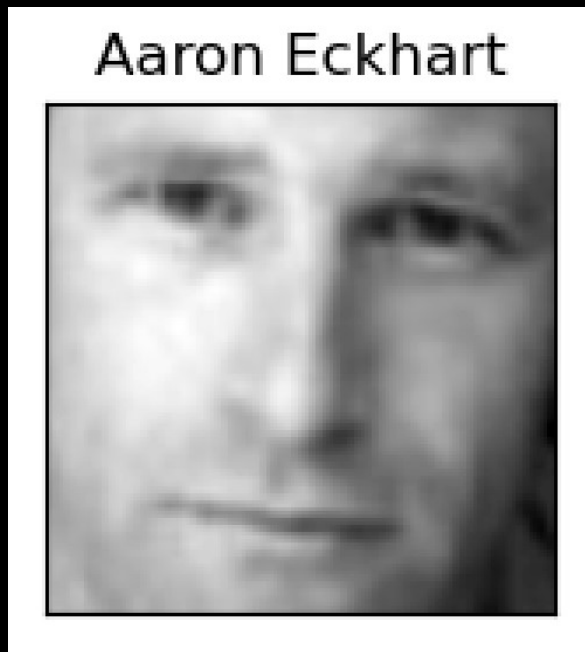
$$\text{Coefficient_stack_ratio_k} = \frac{\sum_{i=0}^k C_i}{\sum_{i=0}^{4095} C_i}$$

Eigenface의 coefficient를 중요한
순서부터 더하고 총합으로
나누어 그 변동을 본 결과,
Coefficient는 1000번째가 넘으면
영향이 미미하므로 eigenface를
1024개, 512개로 줄여본다.

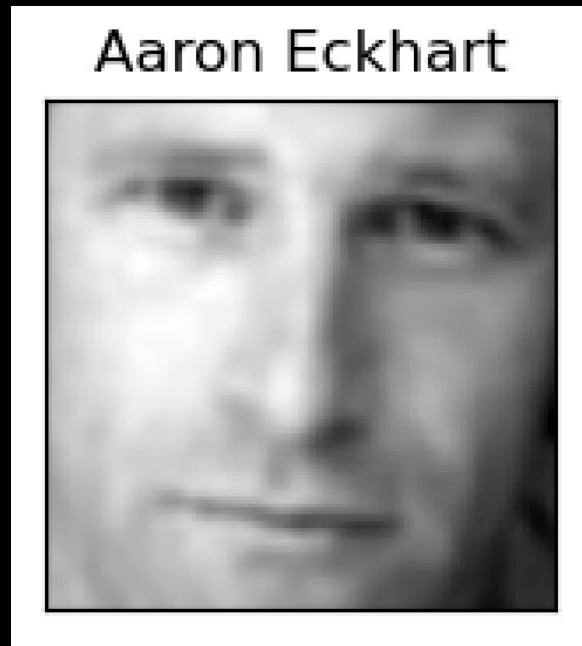


Results – Finding eigenfaces

(1) 12288개의 64x64 grey face image, 1024개, 512개의 eigenface



Original



By 1024 eigenfaces



By 512 eigenfaces

1024개는 거의 비슷한 reconstruction, 512개는 조금 흐려짐

➡ 1024개로 결정

Works – Validation & Test

Validate Set 구성:

인종과 성별 등을 고려하여 서양 여자와 남자 2명씩, 한국 여자와 남자 2명씩, 개(골든 리트리버)와 고양이(흰색)의 사진 10장씩 총 100장으로 이루어져 있음

Test Set 구성:

Validate Set과 동일한 인물들로 사진이 각각 5장씩 총 50장으로 이루어져 있음.

실험 과정:

- (1) Validate Set에서 동일 인물의 사진들 간의 코사인 유사성 계산을 통하여 특징적인 coefficient를 추출한다.
- (2) Test Set으로 인물이 동일한지 판별한다.

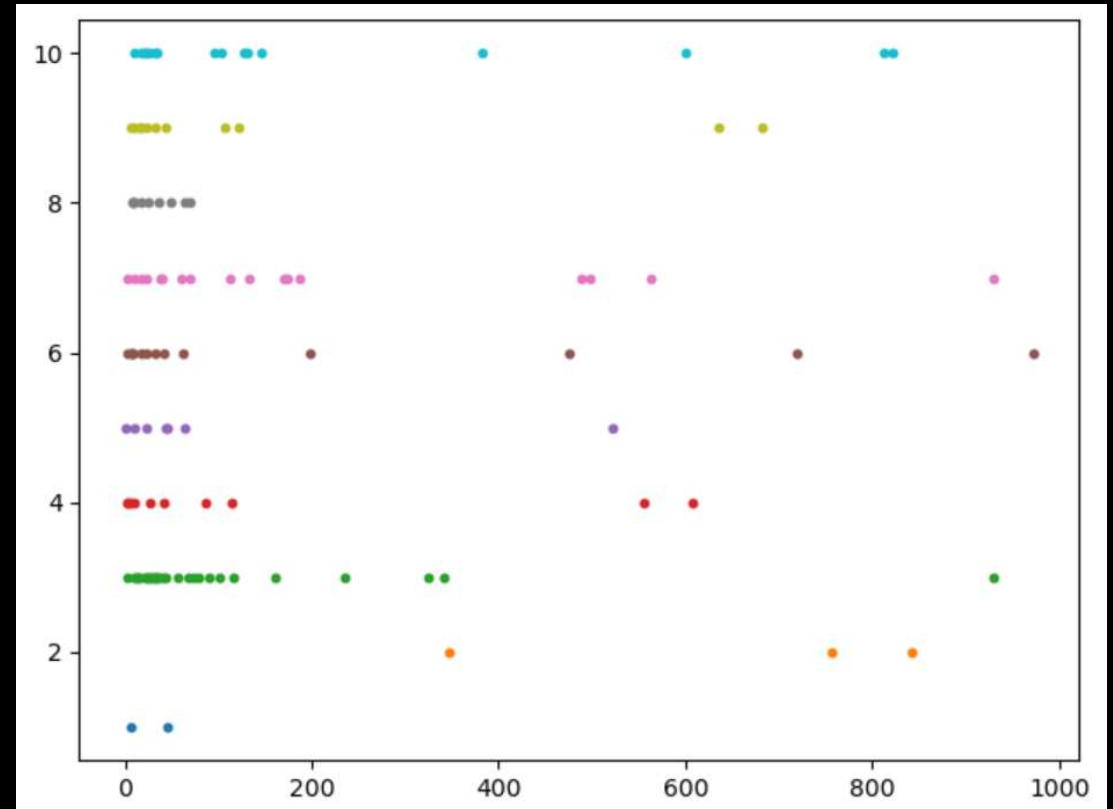
Works – Validation

특징 coefficient 추출:

1024개의 coefficient 중,
동일인 10장 중 2장을
고르는 사진 조합 $10C_2$
=45가지 경우에서 항상
cosine_similarity=1.0이
나오는 coefficient 추출

```
eigenface.py x eigenface.ipynb x
188
189
190 def cos_sim(A, B):
191     return np.dot(A, B) / (np.linalg.norm(A) * np.linalg.norm(B))
192
193 step = range(0, num_photos_person * 10, num_photos_person)
194 unique_weight_point = []
195 for k in step:
196     slice_point = []
197     for slice_start in range(1024 - 1):
198         l = []
199         for i in range(k, k+num_photos_person):
200             for j in range(i + 1, k+num_photos_person):
201                 l.append(cos_sim(weights[i][slice_start:slice_start + 1],
202                                weights[j][slice_start:slice_start+1]))
203         l = np.array(l)
204         if(l.mean() > 0.9):
205             #print("cosine similarity is 1.0 if slice at {}".format(slice_start))
206             slice_point.append(slice_start)
207         unique_weight_point.append(slice_point)
```

Results – Validation

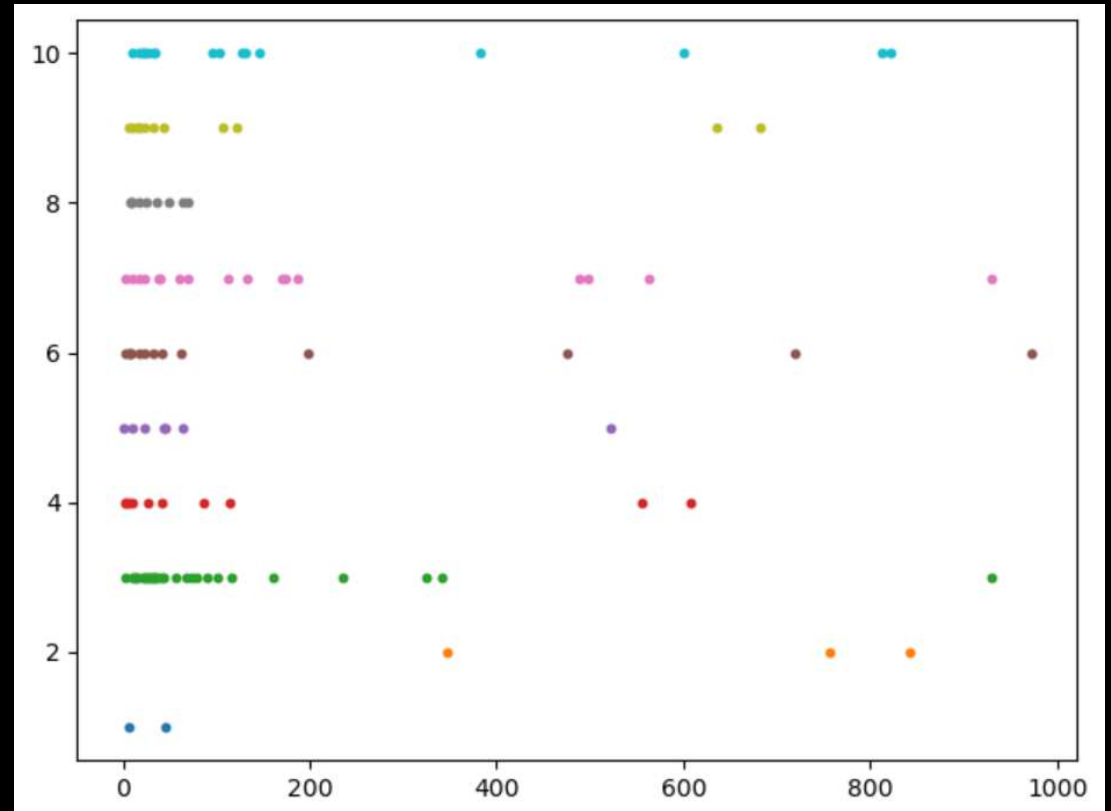


사진은 위에서부터 아래로 1~10번.
1024개의 coefficient중 항상 cosine similarity가 1.0인 점들 표현

Analysis – Validation

대부분 cosine similarity=1.0인 coefficient들은 eigenface number가 100보다 작은 경우에 많이 몰려 있고, 공통 coefficient 조합은 서로 다르다.

하지만 동양, 서양, 남자, 여자, 개, 고양이 중 어느 것도 공통점 차이점을 발견할 수 없었다.



Works – Test

Val



Test



한 장의 Test Image 당 10장 모두의 Validation Image에 대해 특징 coefficient 지점에서 Cosine similarity 시행.
85% 이상의 특징 coefficient 지점에서 Cosine similarity=1.0이 나온다면 일치하는 얼굴이라고 판단.

Results – Test

	1	2	3	4	5	Accurate Rate
Anya Taylor-Joy	False	True	True	True	True	80%
Benedict Cumberbatch	False	False	False	False	False	0%
Cat	True	True	True	True	True	100%
Dog	True	False	False	False	False	20%
Emma Watson	False	True	False	True	True	60%
Jo Yuri	False	False	False	False	True	20%
Kang Dongwon	True	False	False	False	True	40%
Kim Chaewon	False	True	False	True	False	40%
Leonardo Dicaprio	False	False	False	False	False	0%
Yu Ain	False	True	False	False	False	20%

평균 accurate rate는 38%로, 인식률이 많이 낮다.

Analysis – Test

- 인식률이 100% 또는 0%로 나온 경우는 어떤 얼굴을 줘도 True를 내놓거나, 어떤 얼굴을 줘도 False를 내놓는 경우에 가까울 것.
- 따라서 True와 False가 적절히 섞여 있는 경우에서 내놓는 True가 오히려 신뢰성이 높다고 판단.
- 고양이의 경우 하얀 고양이를 사용했기 때문에 눈 코 입 부분을 제외한 곳은 모두 하얗다. 이것이 특징적으로 작용하여 고양이를 인식했을 가능성이 높다.

Results – Test2



	Face Type	Result
Anya Taylor-Joy	Laughing	False
Benedict Cumberbatch	Laughing	False
Cat	Not white	False
Dog	Eyes closed	False
Emma Watson	Laughing	False
Jo Yuri	Laughing	False
Kang Dongwon	Laughing	False
Kim Chaewon	Laughing	False
Leonardo Dicaprio	Laughing	False
Yu Ain	Laughing	False

얼굴 표정이 바뀐 경우 모두 인식하지 못한다.

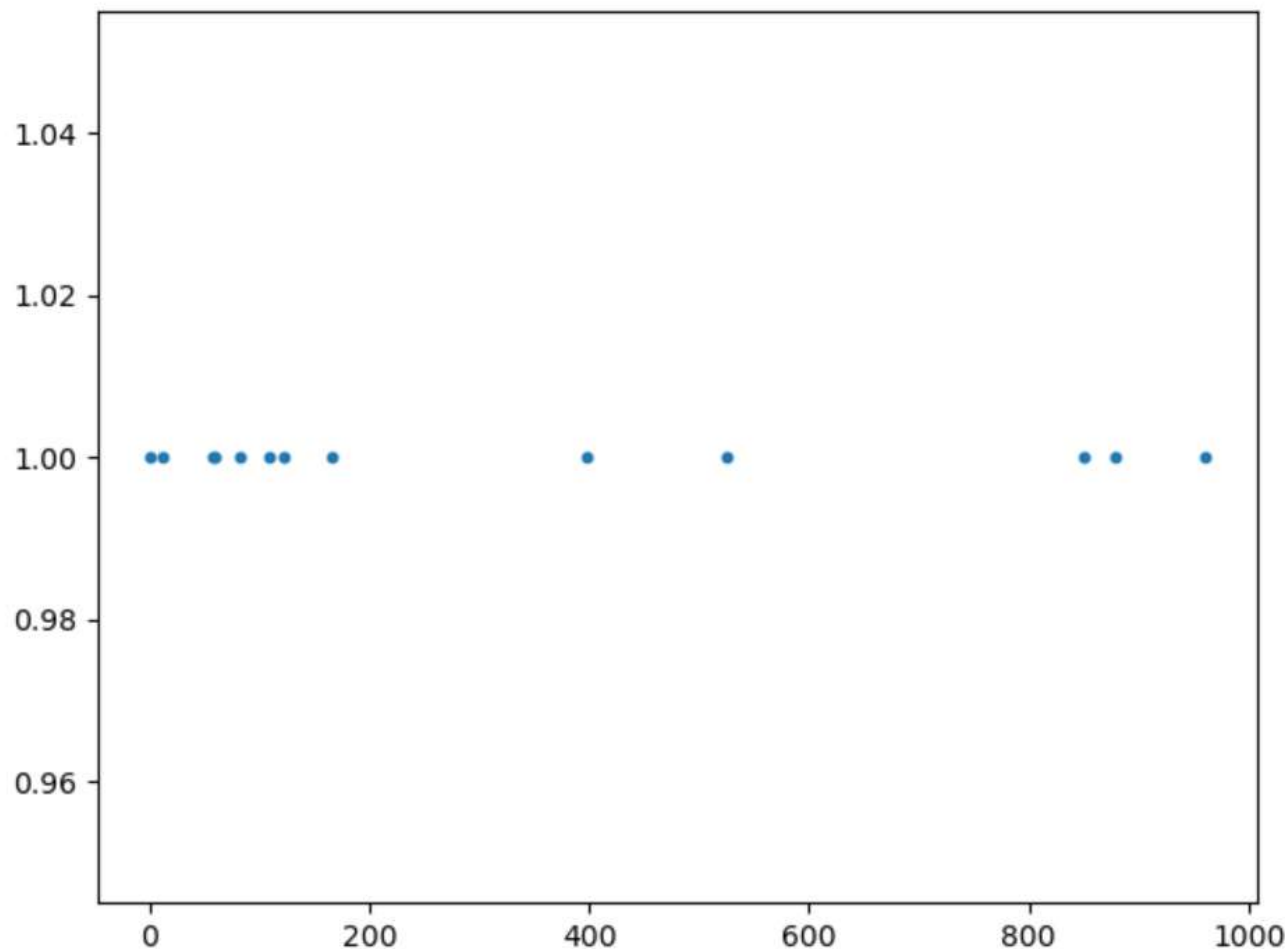
Analysis – Laughing Points

웃는 얼굴들 간의 cosine similarity=1.0인 부분을 추출한 결과 12개의 eigenface가 선택됨

여러가지 eigenface들 중 이 eigenface들이 웃는 얼굴 형성에 중요한 역할을 한다고 생각할 수 있다.

cosine similarity is 1.0 at following points

[0, 12, 57, 59, 82, 109, 122, 166, 397, 524, 851, 879, 960]



Conclusion

- 64x64 크기의 사진에서, 차원의 $\frac{1}{4}$ 인 1024개의 eigenface만으로도 거의 완벽한 재구현을 할 수 있다.
- Cosine similarity를 통해 coefficient를 분석한 결과 각 인물들에 일관적인 방향으로 영향을 미치는 coefficient를 구할 수 있었다.
- 이를 바탕으로 인물을 구별한 결과 38%의 낮은 인식률을 보였다.
- 웃는 얼굴 등 표정의 변화가 생기면 전혀 인식할 수 없었다.