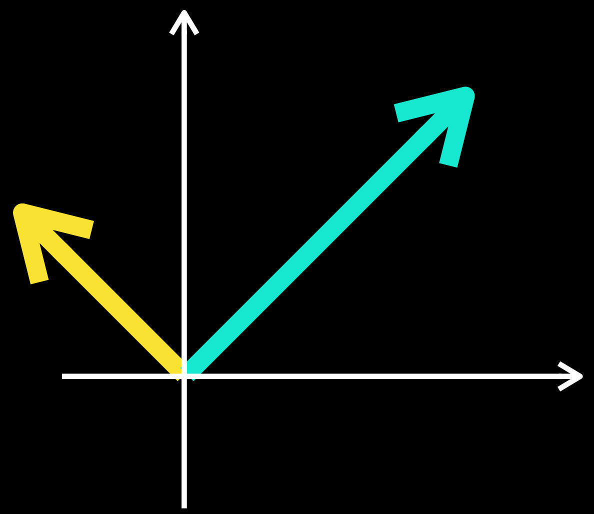
Principal component analysis (PCA) with SVD

Linear Algebra Essentials



PCA

$$cov = \frac{1}{n-1} \sum_{i}^{n} (x_{1i} - \mu_1)(x_{2i} - \mu_2)$$

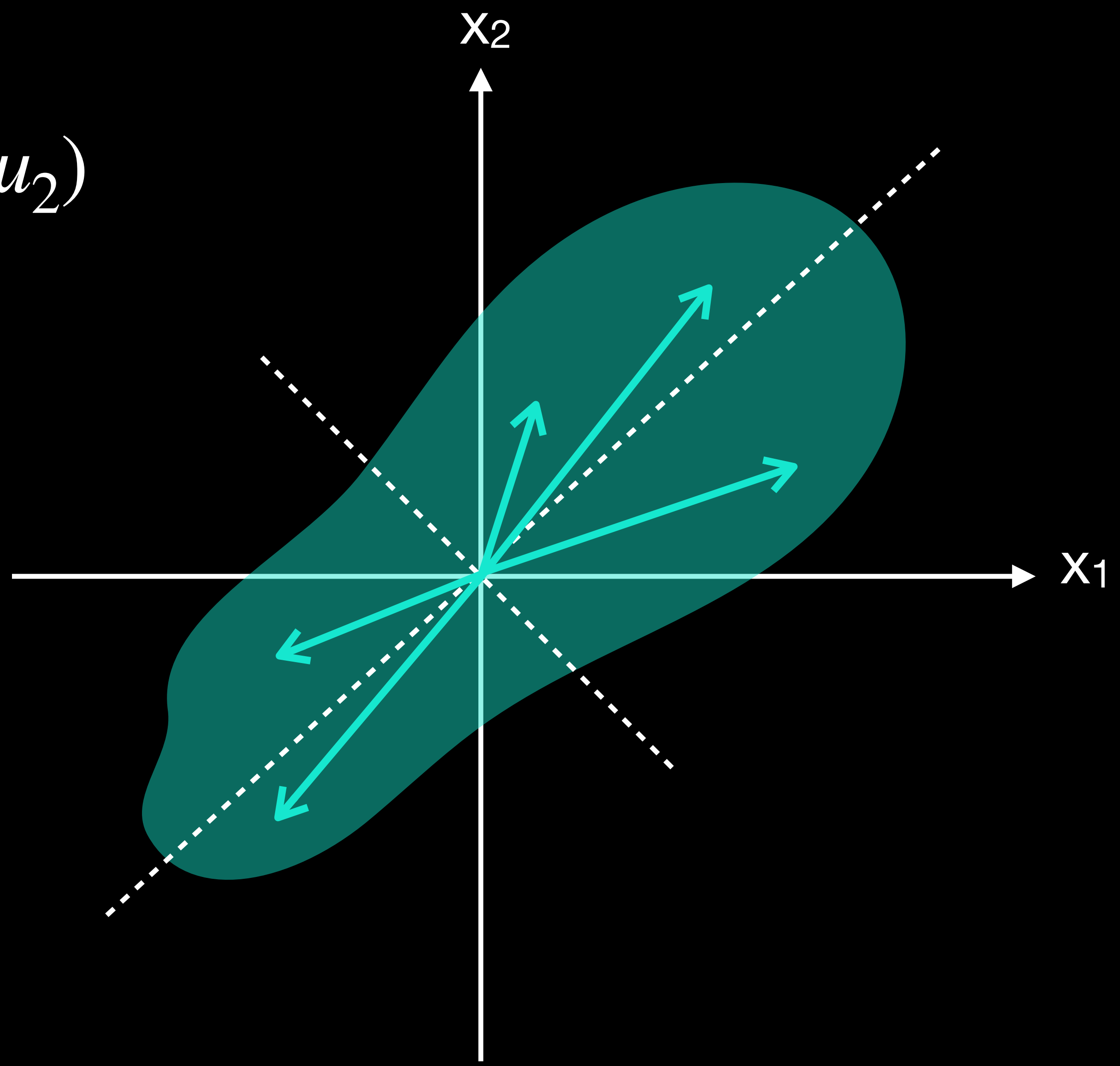
$\Sigma \propto XX^T$ *- must be centered*

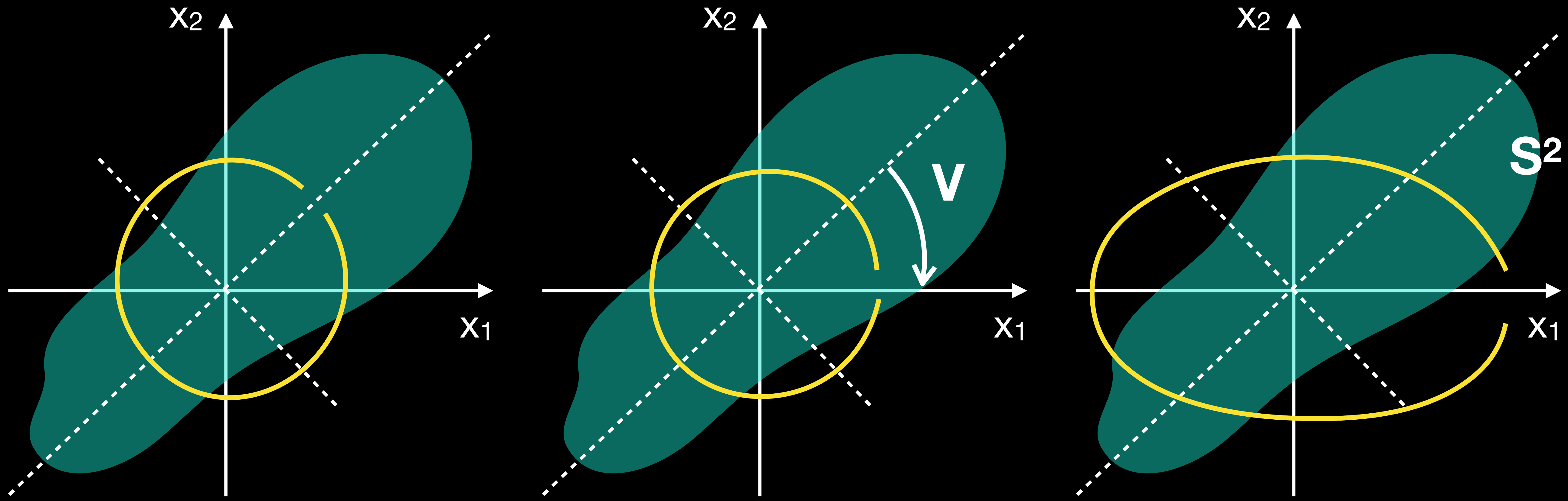
X - (2 x n) matrix

$$X^T = USV$$

$$XX^T = (USV)^T(USV)$$

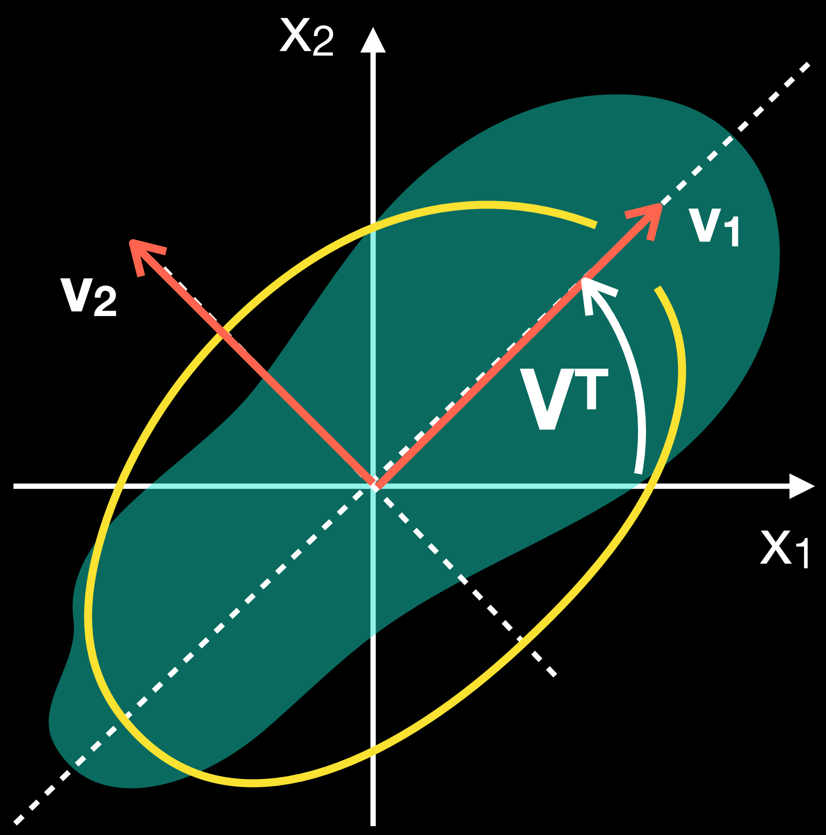
$$= V^T S^T \underbrace{U^T(U}_{=I} SV) = V^T S^2 V$$



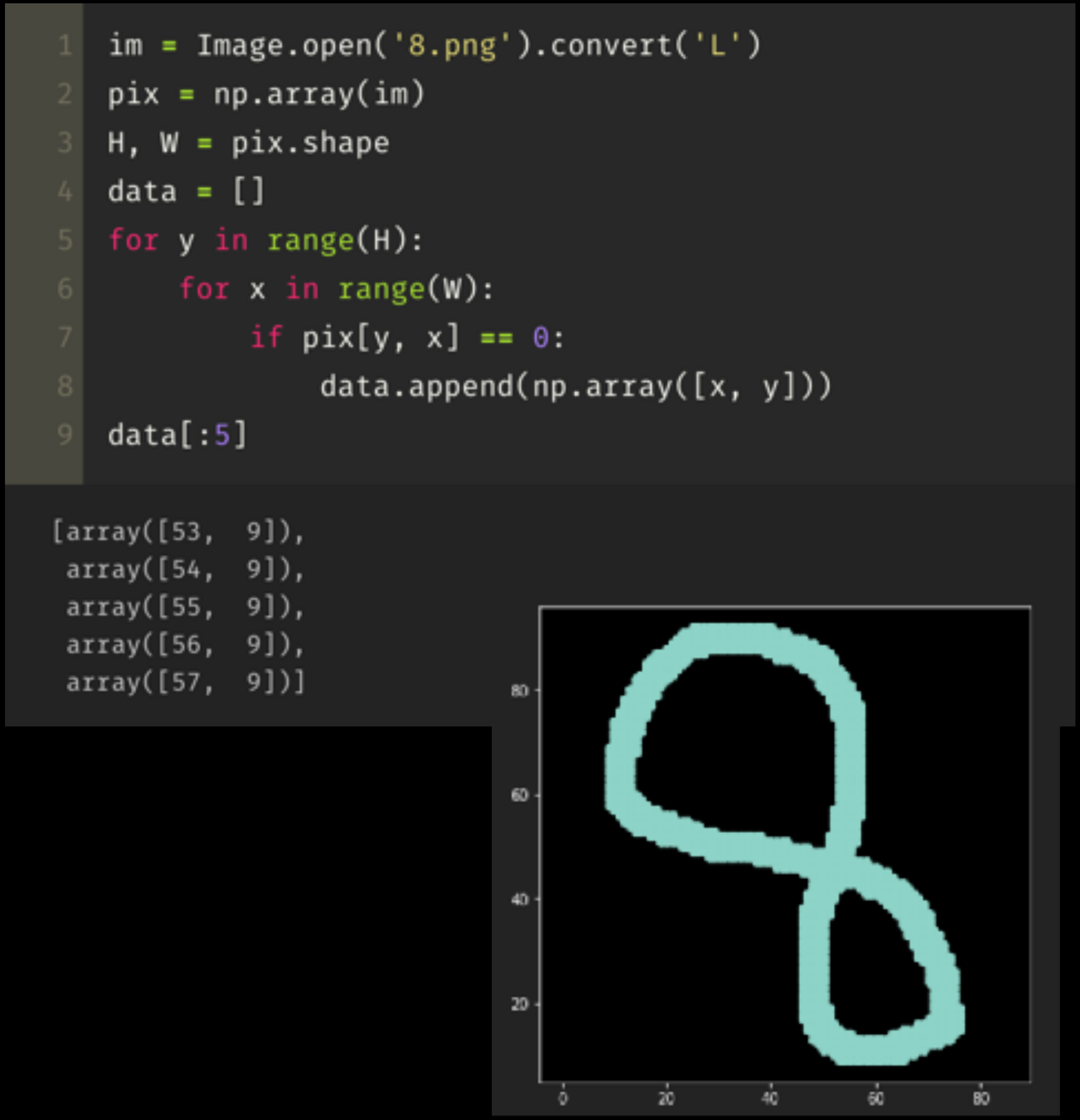


$$XX^T = V^T S^2 V$$

$$V^T = \begin{bmatrix} cos(\alpha) & \text{-}sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} = \left[v_1, v_2 \right]$$

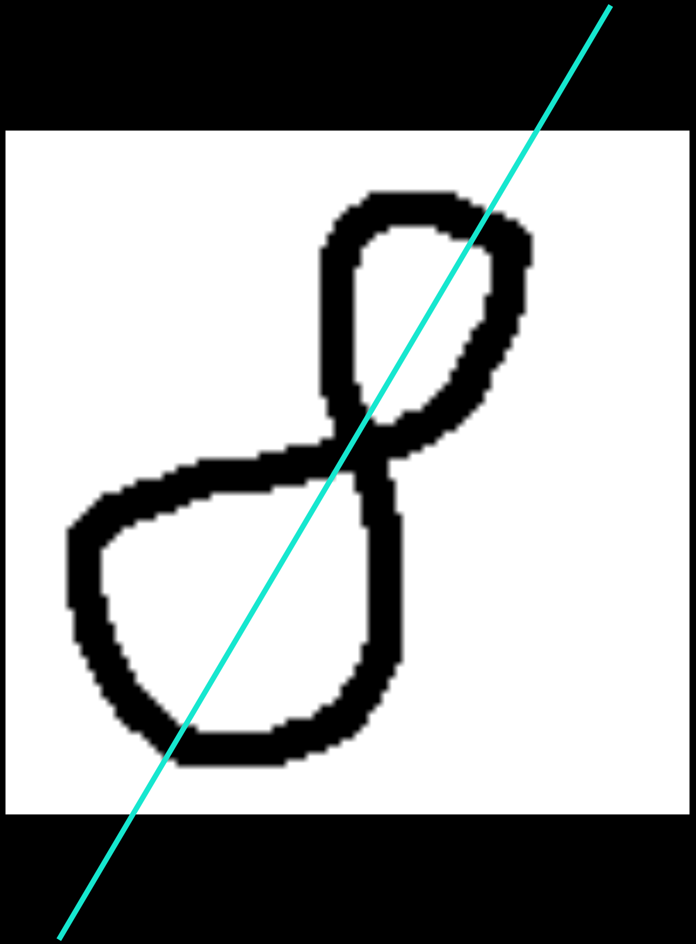


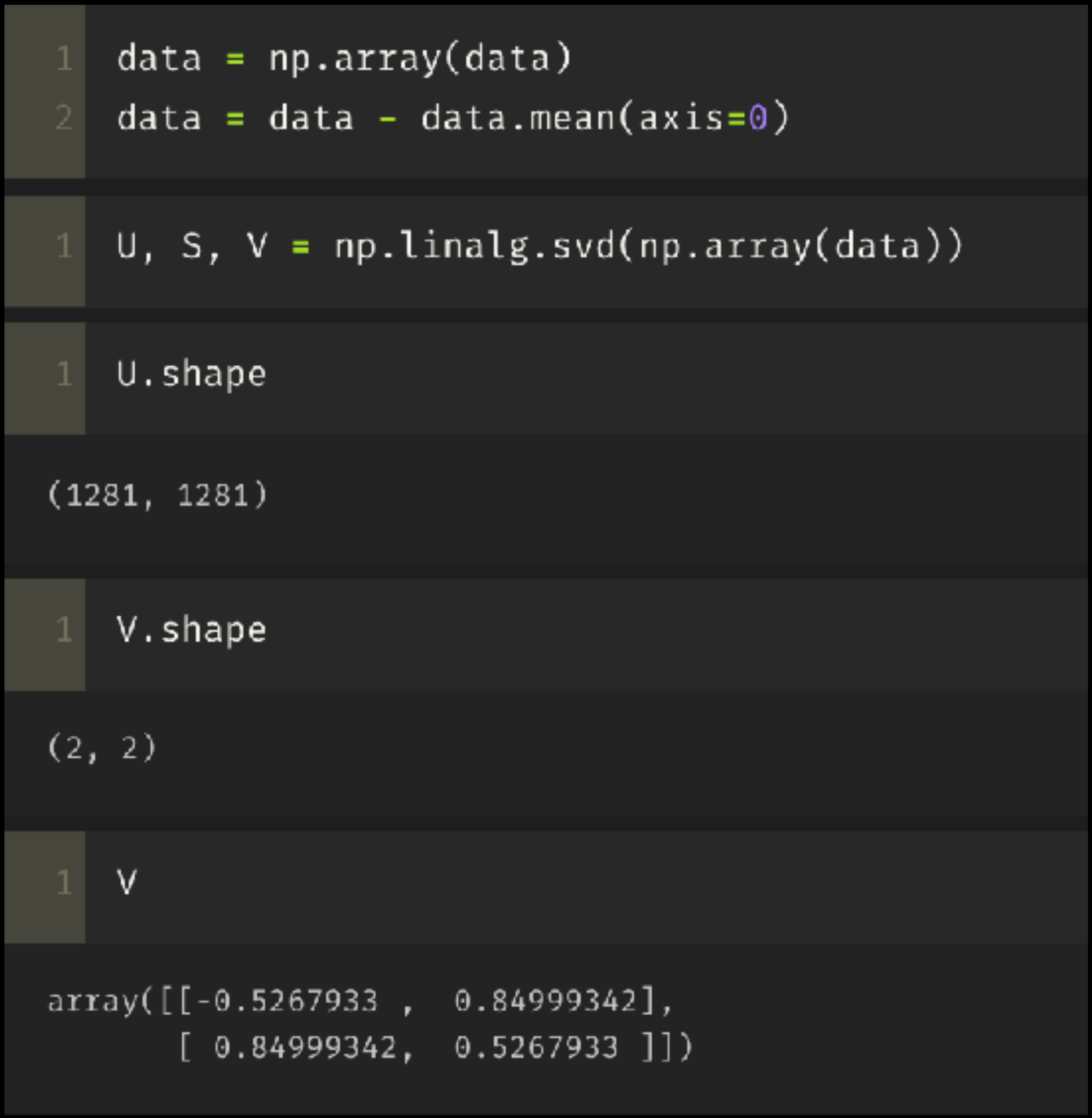
Example: Symbol verticalization



```python
im = Image.open('8.png').convert('L')
pix = np.array(im)
H, W = pix.shape
data = []
for y in range(H):
    for x in range(W):
        if pix[y, x] == 0:
            data.append(np.array([x, y]))
data[:5]
```

```
[array([53,  9]),
 array([54,  9]),
 array([55,  9]),
 array([56,  9]),
 array([57,  9])]
```





```
data = np.array(data)
data = data - data.mean(axis=0)
```

```
U, S, V = np.linalg.svd(np.array(data))
```

```
U.shape
```

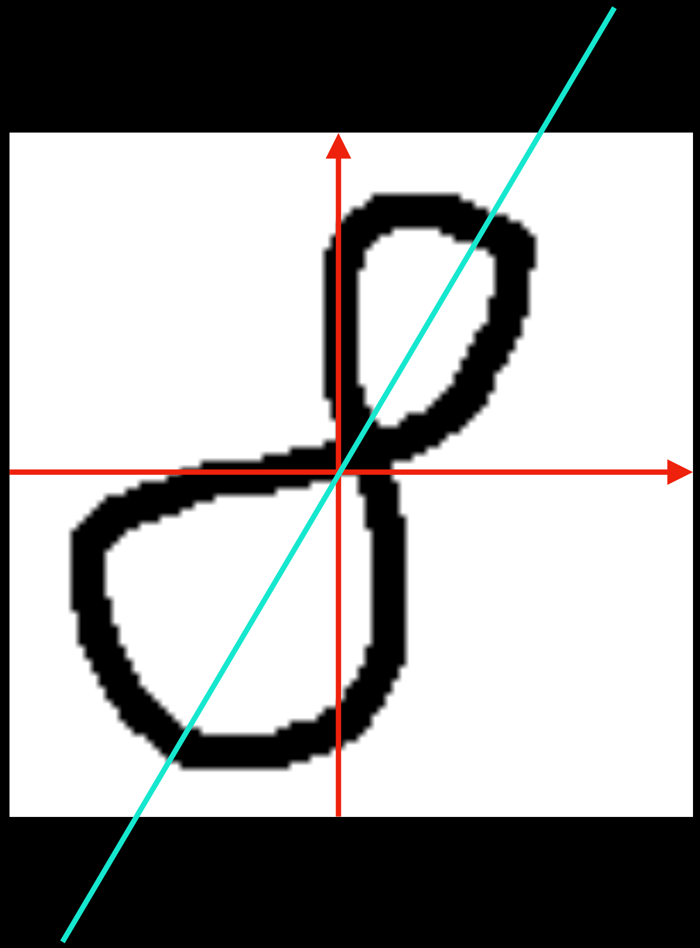
```
(1281, 1281)
```

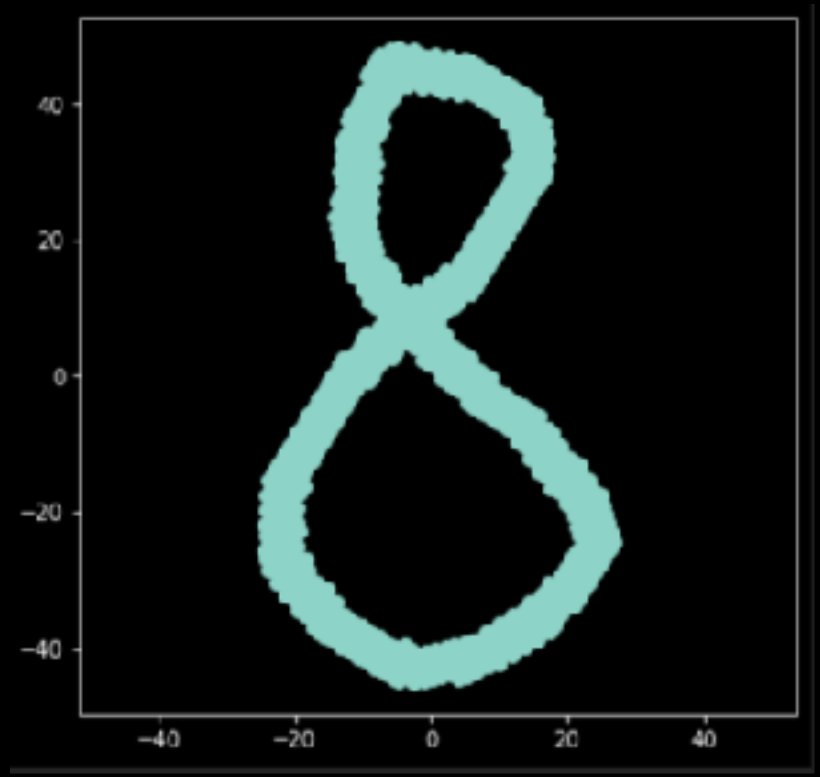
```
V.shape
```

```
(2, 2)
```

```
V
```

```
array([[-0.5267933 ,  0.84999342],
       [ 0.84999342,  0.5267933 ]])
```




40
20
0
−20
−40
−40
−20
0
20
40