

# Face Recognition

Shusen Wang

# Challenge of Face Recognition

**Question:** Why not using Softmax classifier (as in image recognition)?



# Challenge of Face Recognition

**Question:** Why not using Softmax classifier (as in image recognition)?

- Softmax classifier is a dense layer with Softmax activation.
- #class is large (can be millions or billions).
- #parameters in the output layer is huge!
  - Suppose the input shape of Softmax classifier is  $1K$ .
  - Suppose #class is  $10M$ .
  - Then #parameters =  $1K \times 10M = 10G$ .

# Training

# Randomly Sample a Triplet

positive  $\mathbf{x}^+$



anchor  $\mathbf{x}^a$



negative  $\mathbf{x}^-$



# Feature Extraction using CNN

positive

$\mathbf{x}^+$



$\mathbf{f}$



$\mathbf{f}(\mathbf{x}^+)$

anchor

$\mathbf{x}^a$



$\mathbf{f}$



$\mathbf{f}(\mathbf{x}^a)$

negative

$\mathbf{x}^-$

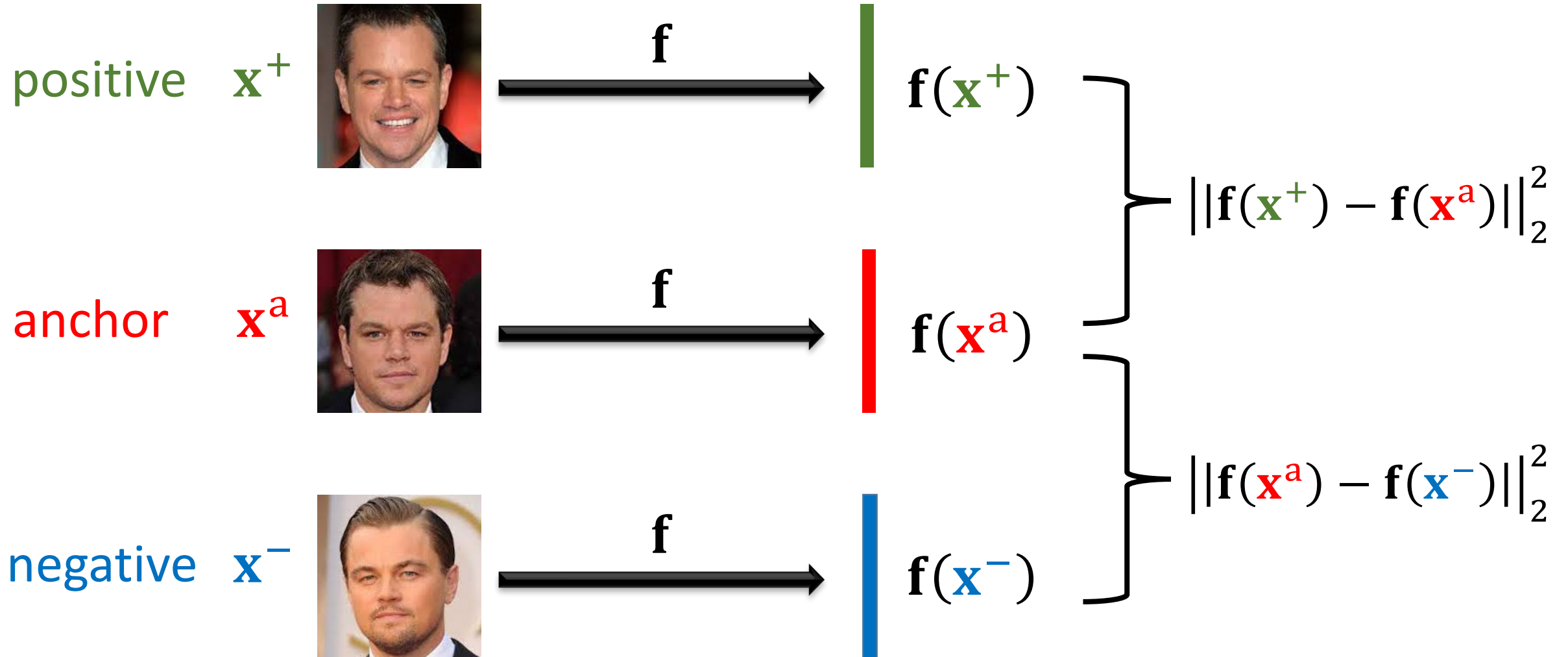


$\mathbf{f}$



$\mathbf{f}(\mathbf{x}^-)$

# Loss Functions



# Loss Functions

- $N$  triplets:  $(\mathbf{x}_1^a, \mathbf{x}_1^+, \mathbf{x}_1^-), (\mathbf{x}_2^a, \mathbf{x}_2^+, \mathbf{x}_2^-), \dots, (\mathbf{x}_N^a, \mathbf{x}_N^+, \mathbf{x}_N^-)$ .
- $N$  can be much larger than  $n$  (#samples).

- Optimization model:

$$\min \frac{1}{N} \sum_{i=1}^N \left[ \|\mathbf{f}(\mathbf{x}_i^+) - \mathbf{f}(\mathbf{x}_i^a)\|_2^2 - \|\mathbf{f}(\mathbf{x}_i^a) - \mathbf{f}(\mathbf{x}_i^-)\|_2^2 \right].$$



# Loss Functions

- $N$  triplets:  $(\mathbf{x}_1^a, \mathbf{x}_1^+, \mathbf{x}_1^-), (\mathbf{x}_2^a, \mathbf{x}_2^+, \mathbf{x}_2^-), \dots, (\mathbf{x}_N^a, \mathbf{x}_N^+, \mathbf{x}_N^-)$ .
- $N$  can be much larger than  $n$  (#samples).

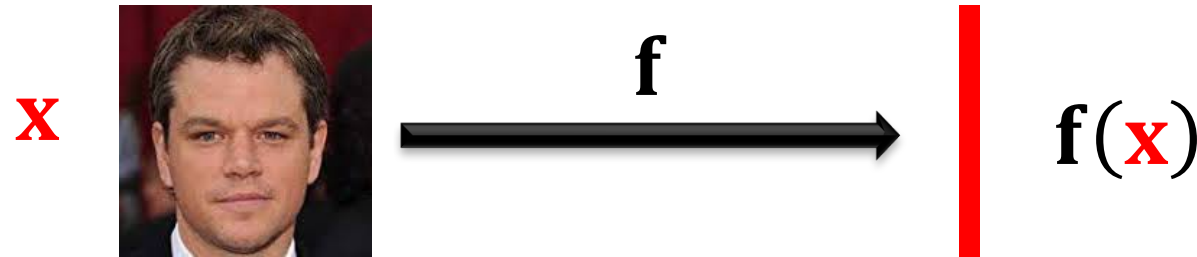
- Optimization model:

$$\min \frac{1}{N} \sum_{i=1}^N \left[ \|\mathbf{f}(\mathbf{x}_i^+) - \mathbf{f}(\mathbf{x}_i^a)\|_2^2 - \|\mathbf{f}(\mathbf{x}_i^a) - \mathbf{f}(\mathbf{x}_i^-)\|_2^2 + \alpha \right]_+.$$

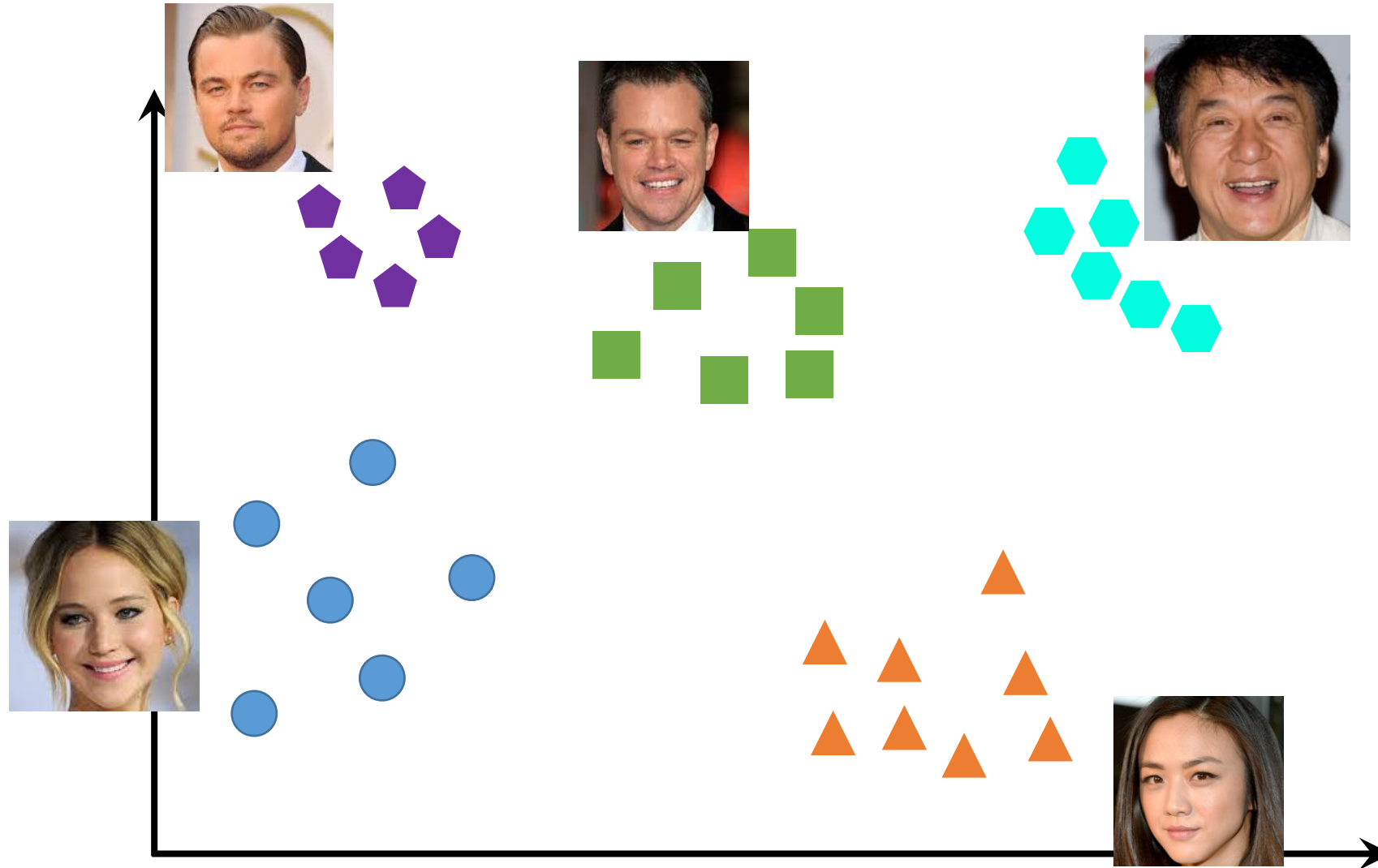
- Here,  $\alpha > 0$ , and  $[z]_+ = \max\{z, 0\}$ .

**Prediction**

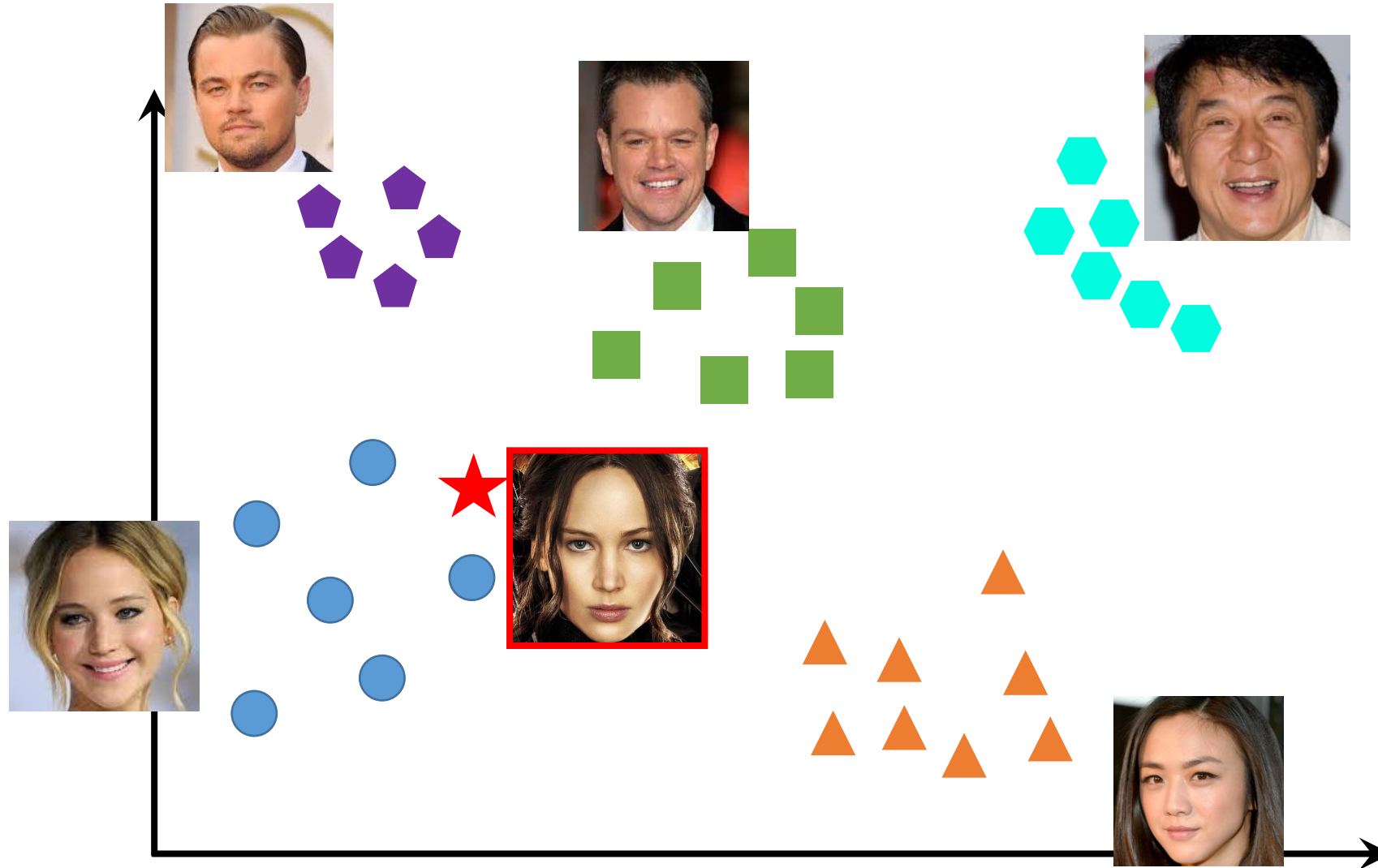
# 1. Feature Extraction for All the **Training** Photos



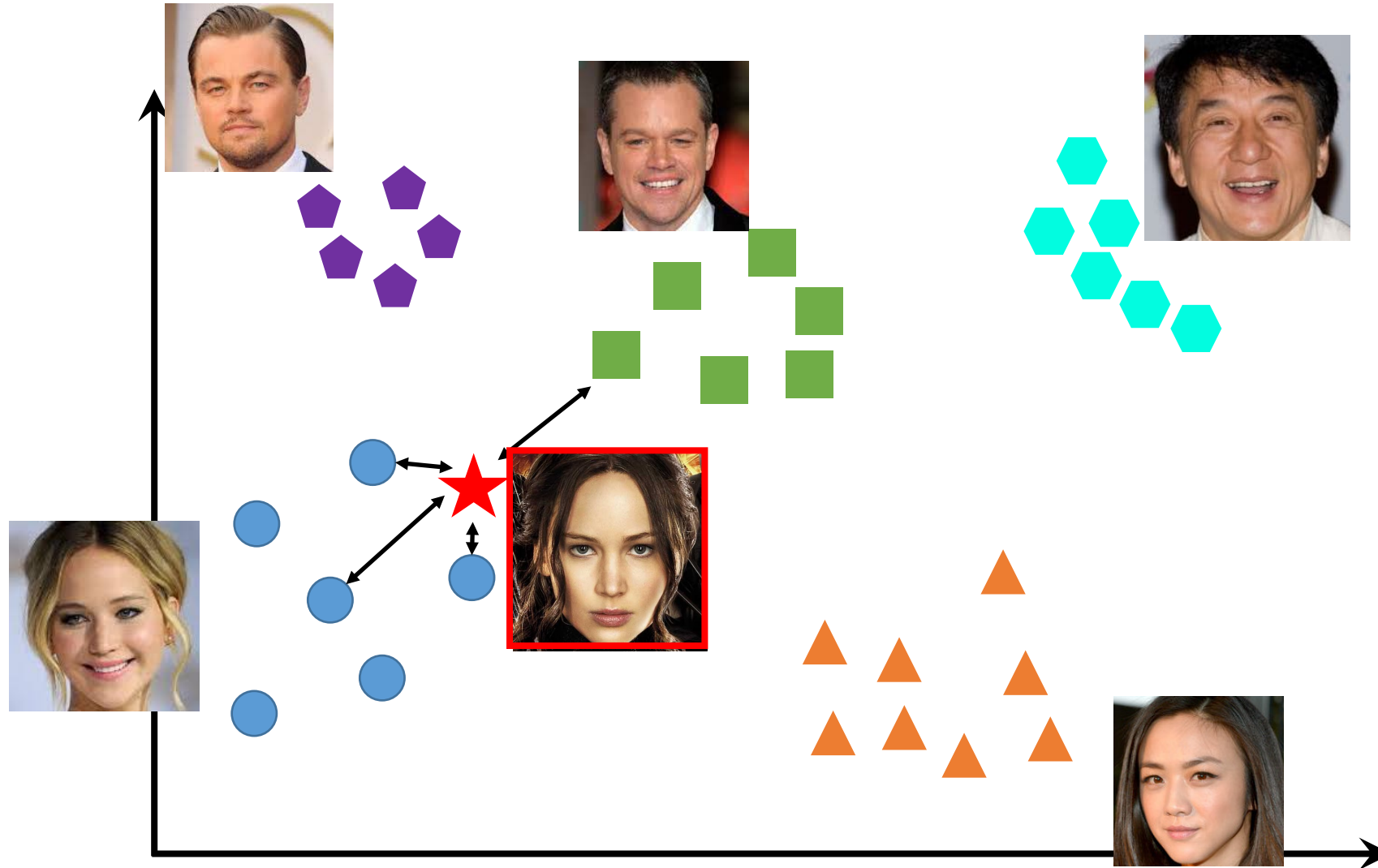
# 1. Feature Extraction for All the **Training** Photos



## 2. Feature Extraction for **Test** Sample



# 3. KNN Classifier



# Keras Implementation

- <http://krasserm.github.io/2018/02/07/deep-face-recognition/>