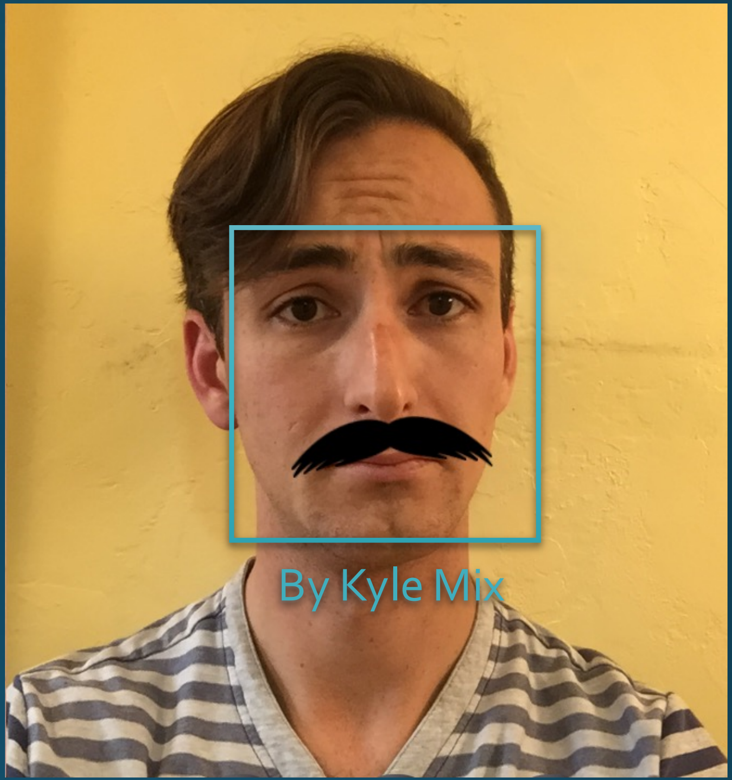


Computer Vision:

Methods for facial recognition

Images = Matrices

Pixels Wide x Pixels High x Channels Deep

Image Dimensionality:

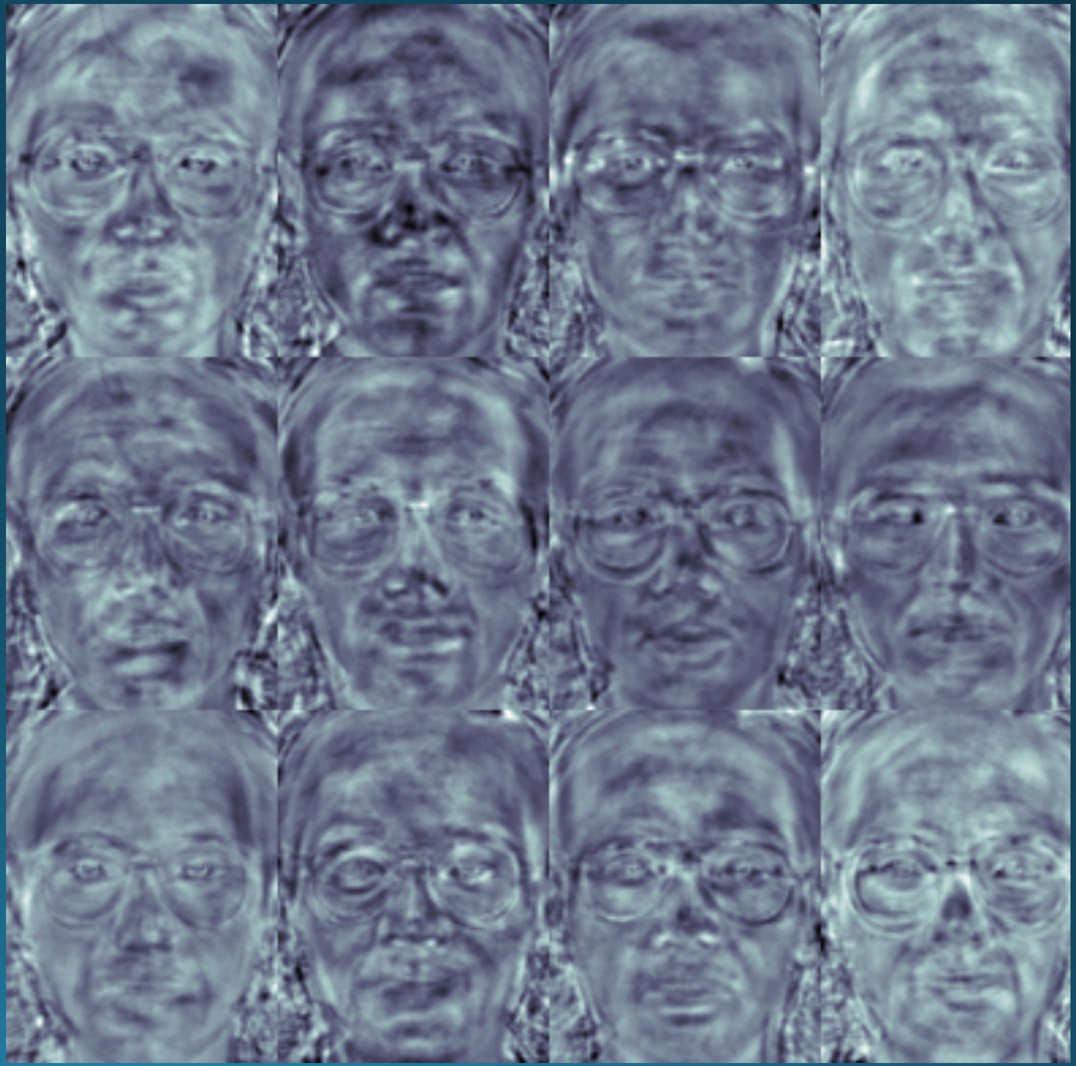
Real world challenges: facial identification

- Expressions, hair, glasses, angle etc. all change
- Images captured under ∞ conditions

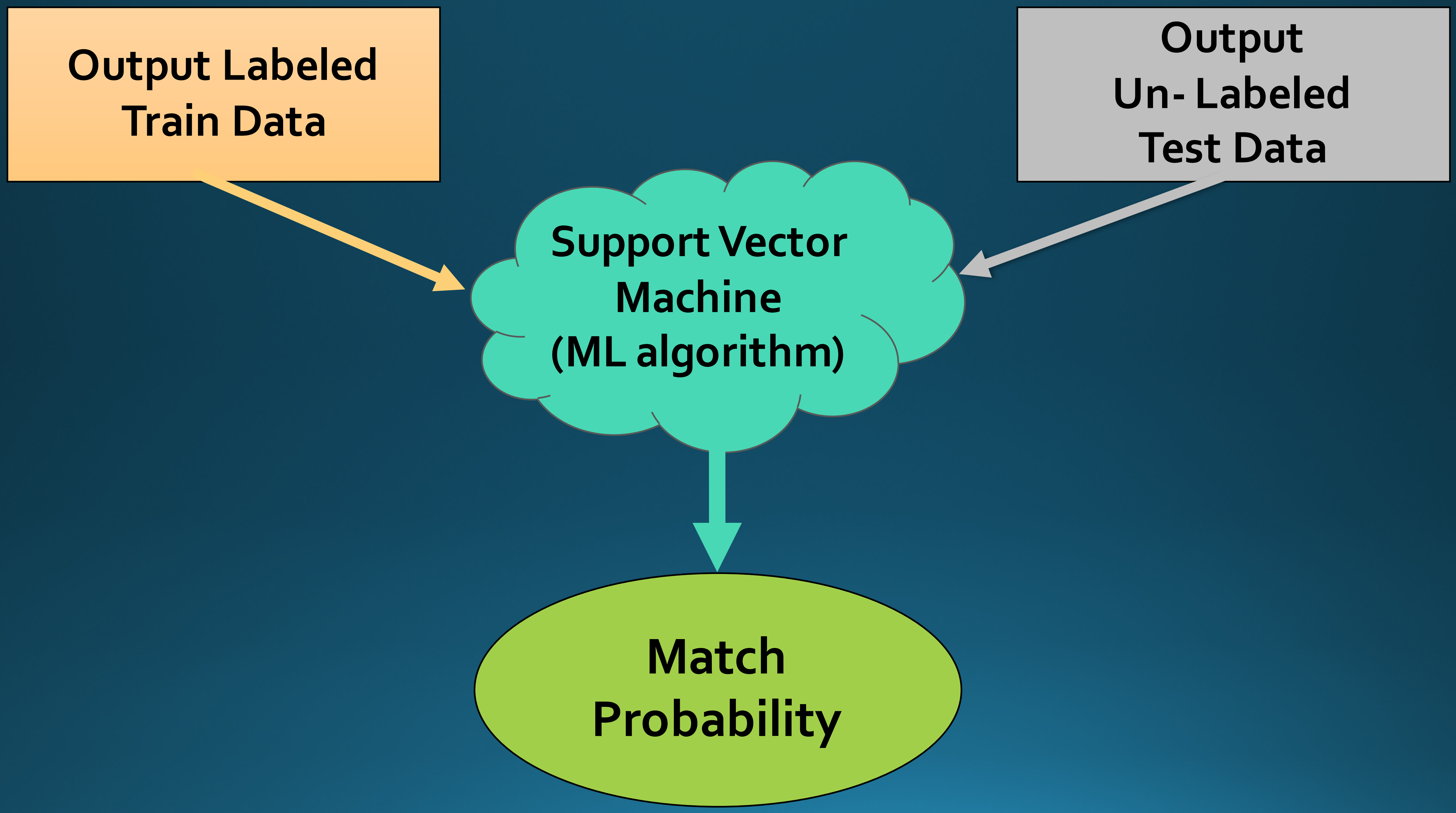
“Vanilla” comparison (Correlation model) = extremely computationally & memory intensive

EigenFaces Method

FisherFaces Method



EigenFaces Method		FisherFaces Method
Input image matrix	High Dim. Input	Input image matrix
PCA Reduce Dimension Max. Variance	↓	Reduce Dimension Max. Variance
		Max. class separation Min. in class scatter
"EigenFaces" output matrix	Low Dim. Output	"FisherFaces" output matrix


Output Labeled Train Data
Output Un- Labeled Test Data
Support Vector Machine (ML algorithm)
Match Probability

EigenFaces

Advantages

- Fast
- Effective with small training sets
- Easy to implement
- Very accurate under ideal conditions

EigenFaces

Disadvantages

- Error prone when conditions change
- Very sensitive to scale
- Can classify on unrelated but valid sources of variation

FisherFaces

Advantages

- More robust when dealing with varying conditions
- Designed to be independent of lighting
- Better at dealing with expressions

FisherFaces

Disadvantages

- More complicated to implement
- Can be challenging to optimize

In conclusion:

- Both algorithms are remarkably effective and well studied.
- Situation and data will dictate which is best
- FisherFaces has been shown to achieve generally lower error rates then alternative methods.

Resources:

- Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection
  - http://www.cs.columbia.edu/~belhumeur/journal/fisherface-pami97.pdf
- SciKit Learn EigenFace example
  - http://scikit-learn.org/stable/auto_examples/applications/face_recognition.html
- Excellent detailed explanation and implementation of FisherFaces in python
  - http://www.bytefish.de/blog/fisherfaces/
- Linear Digressions podcast, High level accessible explanation
  - http://lineardigressions.com/episodes/2016/2/23/facial-recognition-with-eigenfaces
  - http://lineardigressions.com/episodes/2016/2/23/better-facial-recognition-with-fisherfaces

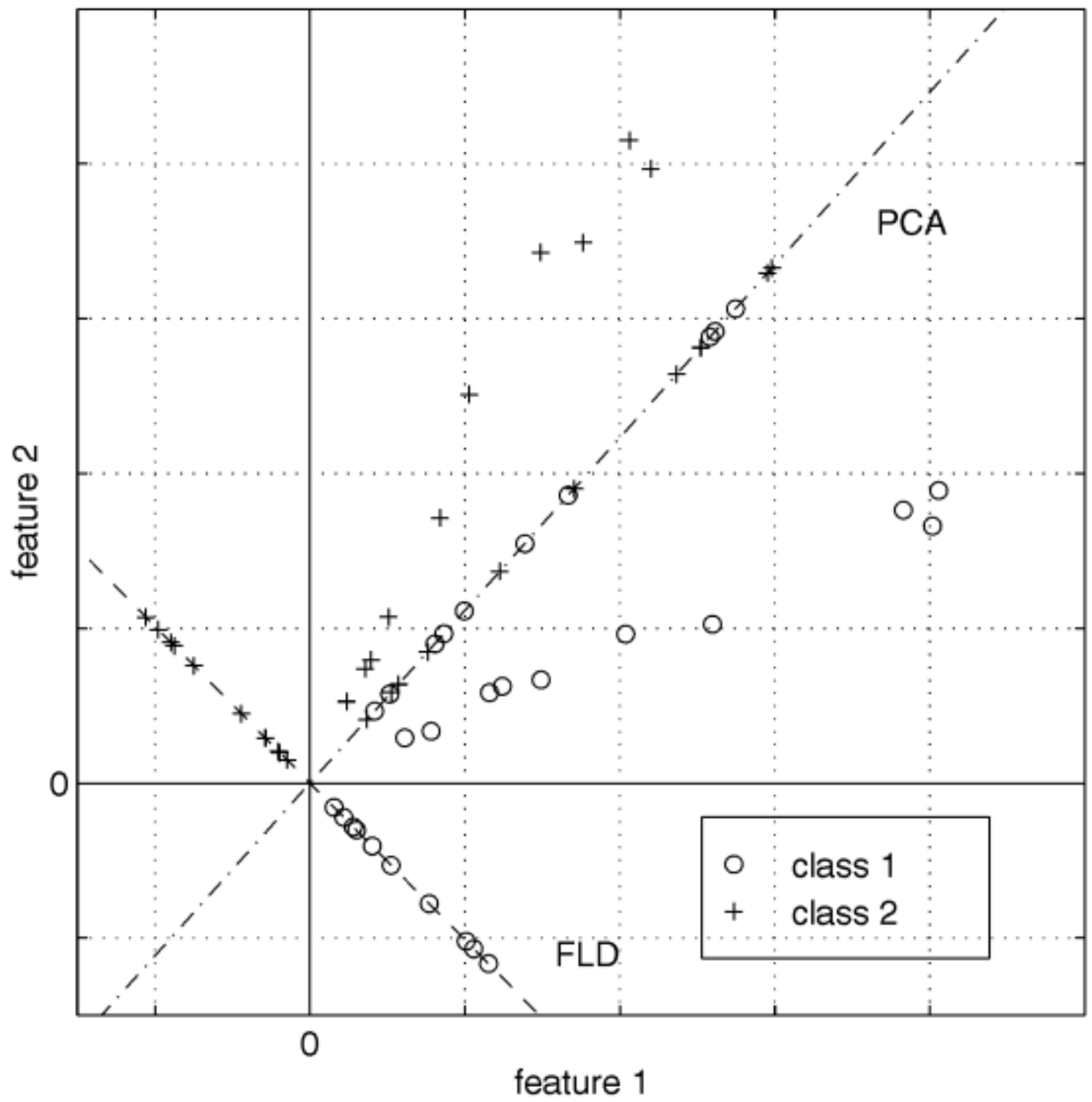
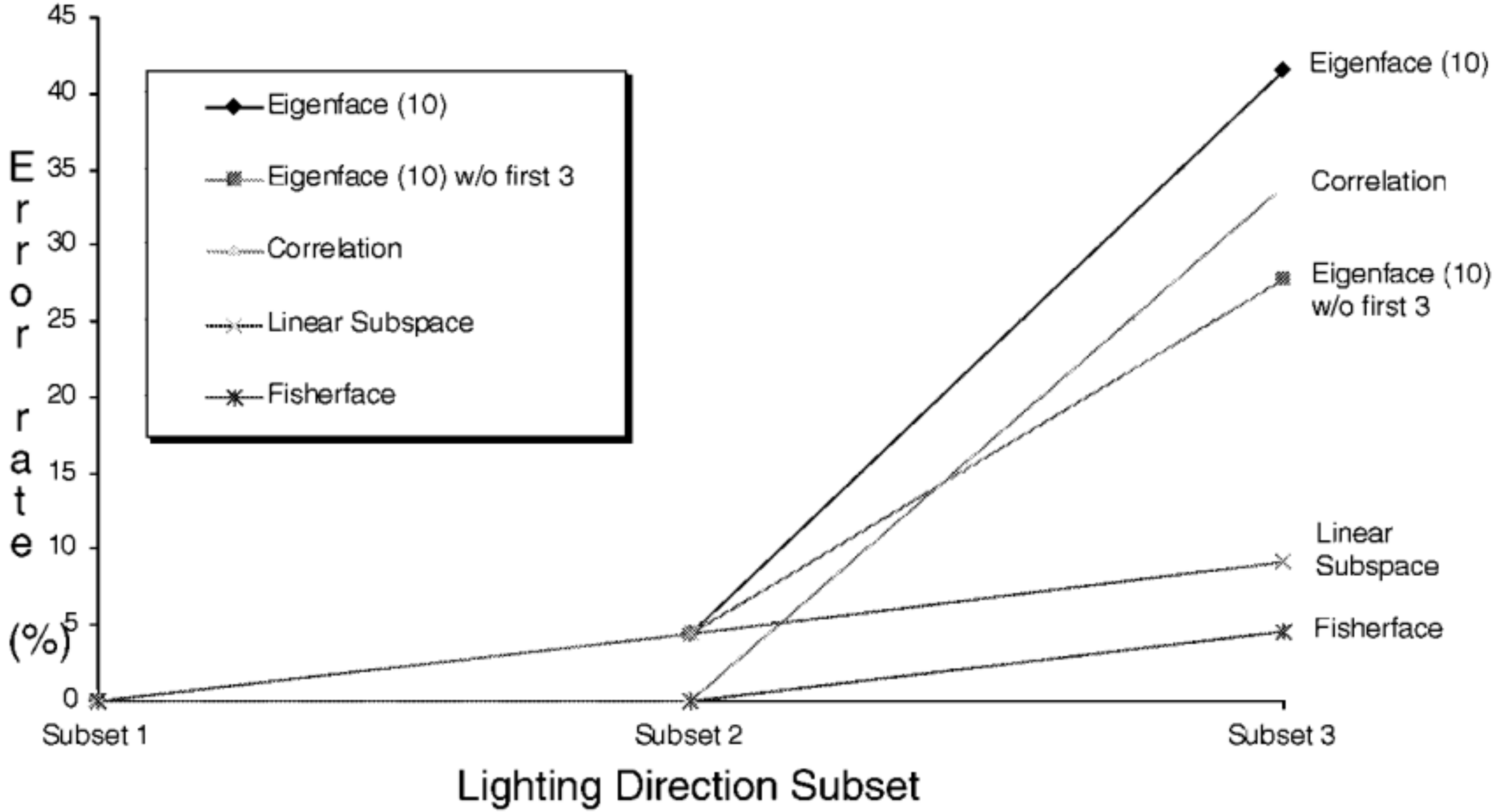
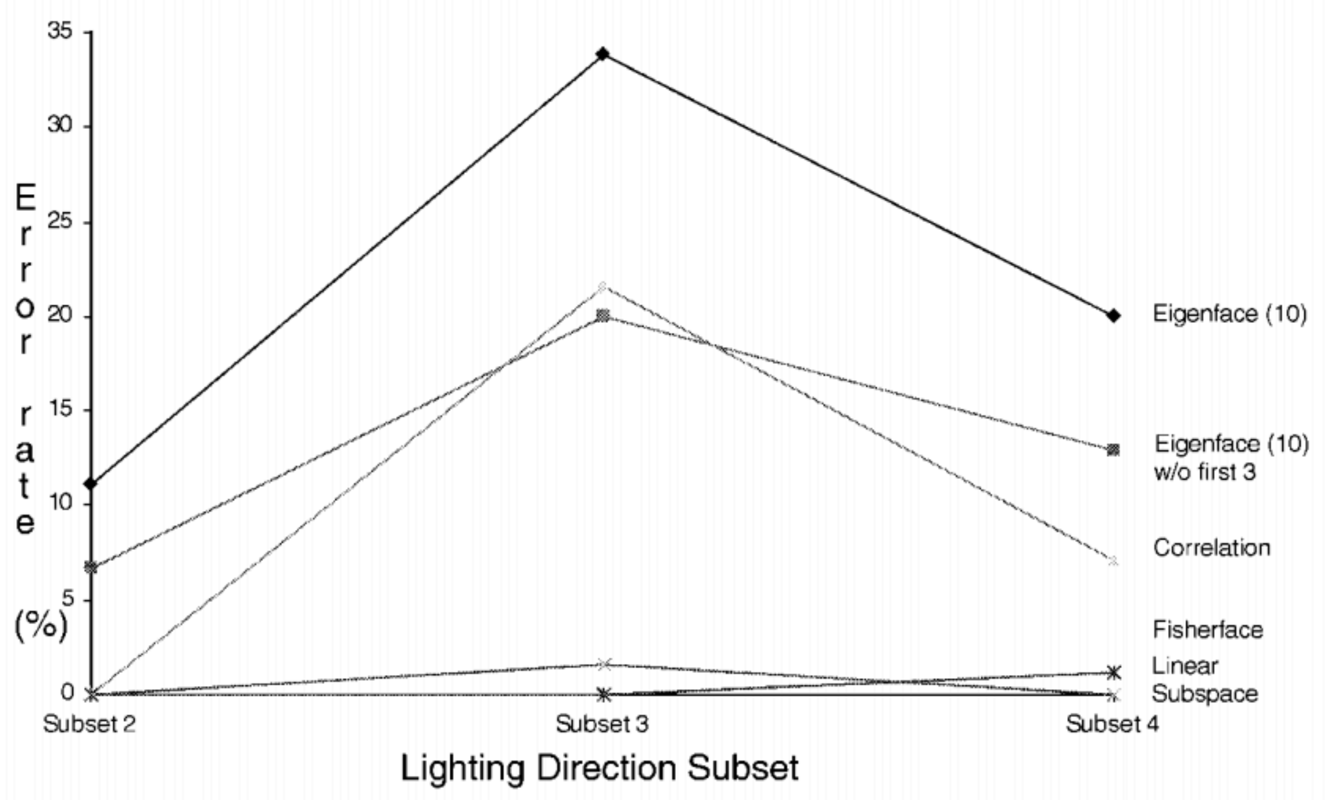


Fig. 2. A comparison of principal component analysis (PCA) and Fisher's linear discriminant (FLD) for a two class problem where data for each class lies near a linear subspace.



45
40
35
30
25
20
15
10
5
0
Error rate (%)
Eigenface (10)
Eigenface (10) w/o first 3
Correlation
Linear Subspace
Fisherface
Eigenface (10)
Correlation
Eigenface (10) w/o first 3
Linear Subspace
Fisherface
Subset 1
Subset 2
Subset 3
Lighting Direction Subset



Interpolating between Subsets 1 and 5				
Method	Reduced Space	Error Rate (%)		
		Subset 2	Subset 3	Subset 4
Eigenface	4 10	53.3 11.11	75.4 33.9	52.9 20.0
Eigenface w/o 1st 3	4 10	31.11 6.7	60.0 20.0	29.4 12.9
Correlation	129	0.0	21.54	7.1
Linear Subspace	15	0.0	1.5	0.0
Fisherface	4	0.0	0.0	1.2

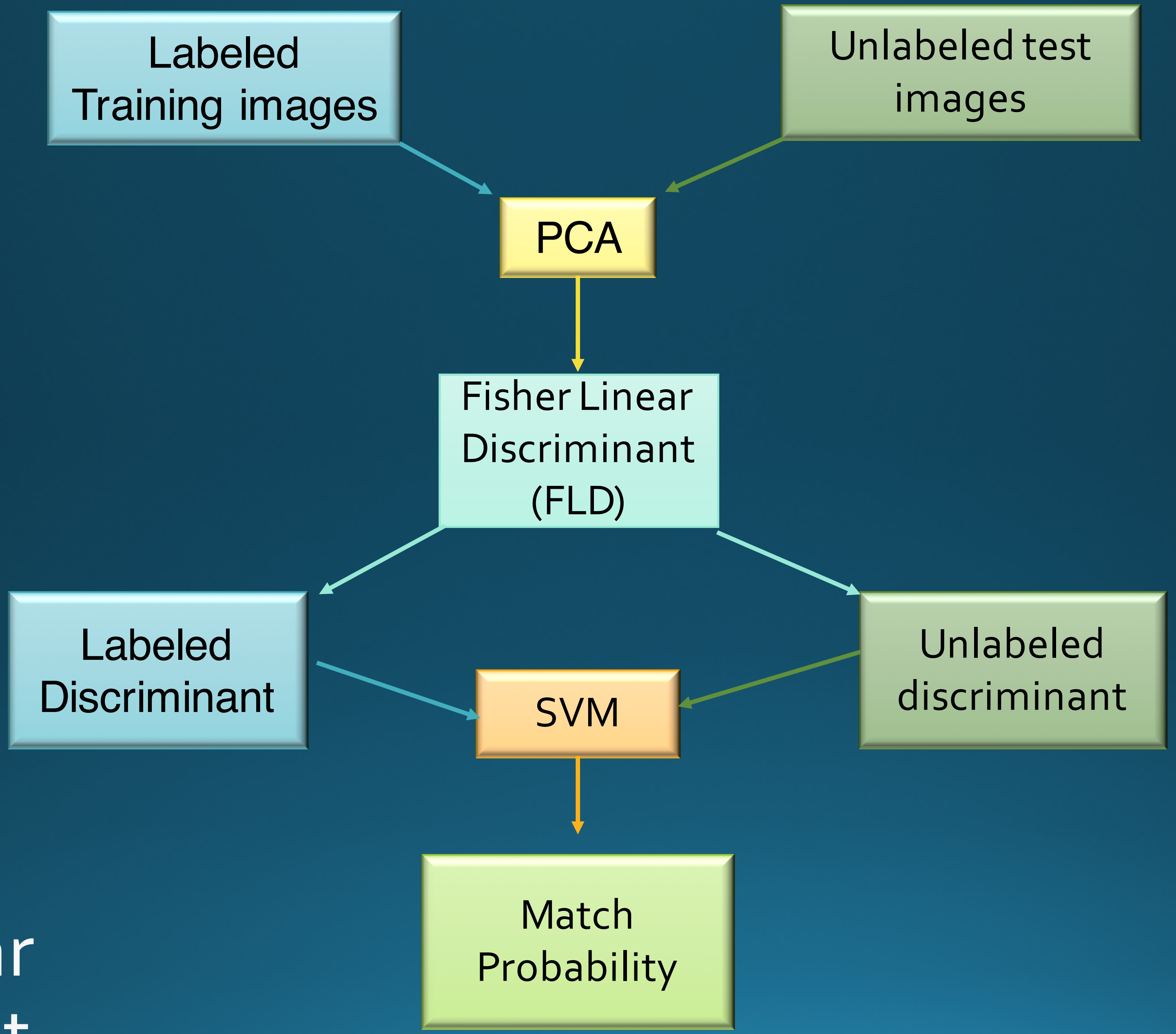
Fig. 6. Interpolation: When each of the methods is trained on images from both near frontal and extreme lighting (Subsets 1 and 5), the graph and corresponding table show the relative performance under intermediate lighting conditions.

Principle component analysis

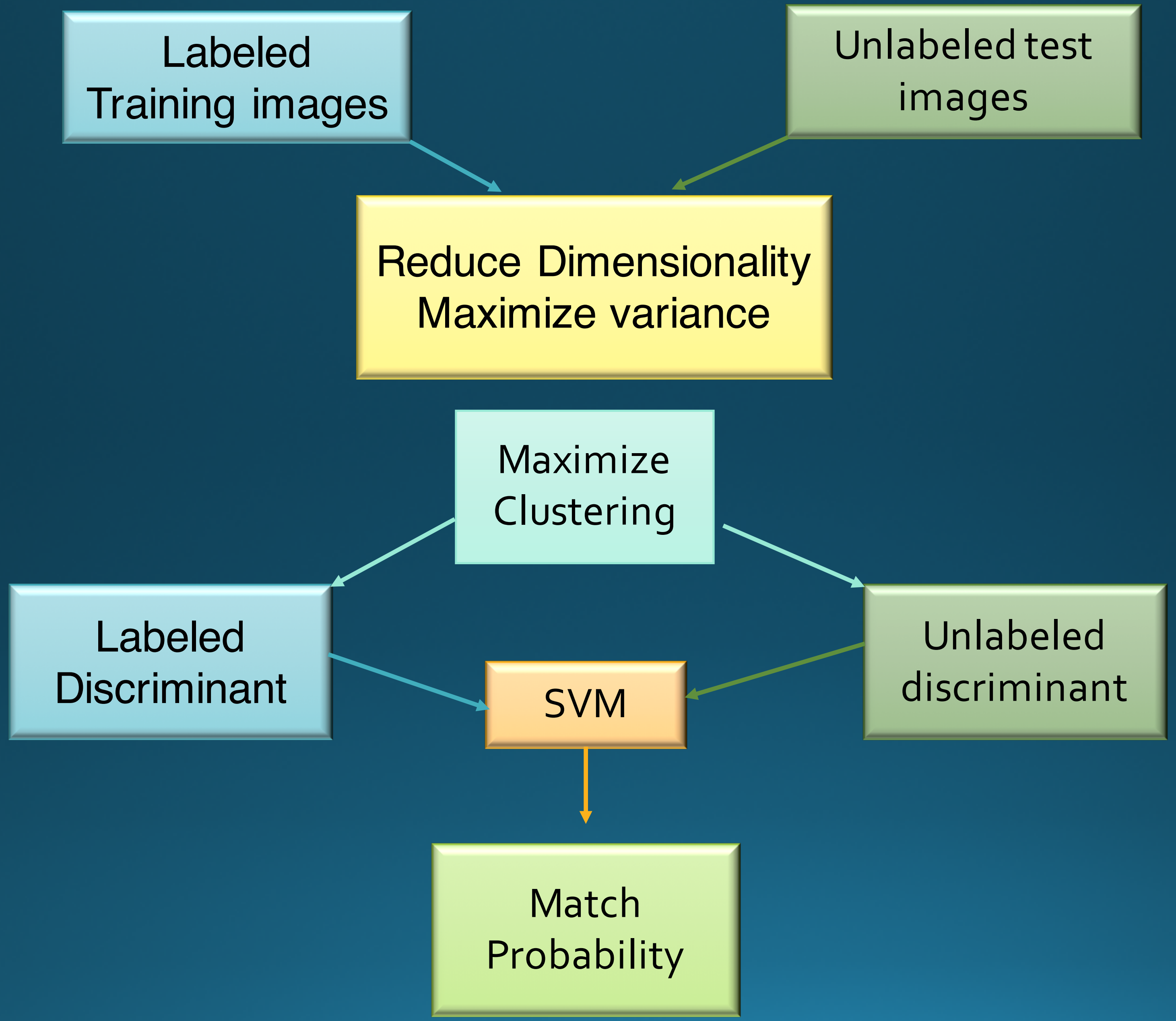
- Orthogonal transformation to convert possibly correlated variables into sets of linearly uncorrelated variables called principal components.
- First principal component accounts for as much of the variance in the data as possible.
- Reveals internal structure of data in a way that best explains it’s variance.
- Supplies a lower-dimensional picture of data when seen from its most informative viewpoint
- Does not take into account any difference in class

Fisher Linear Discriminant

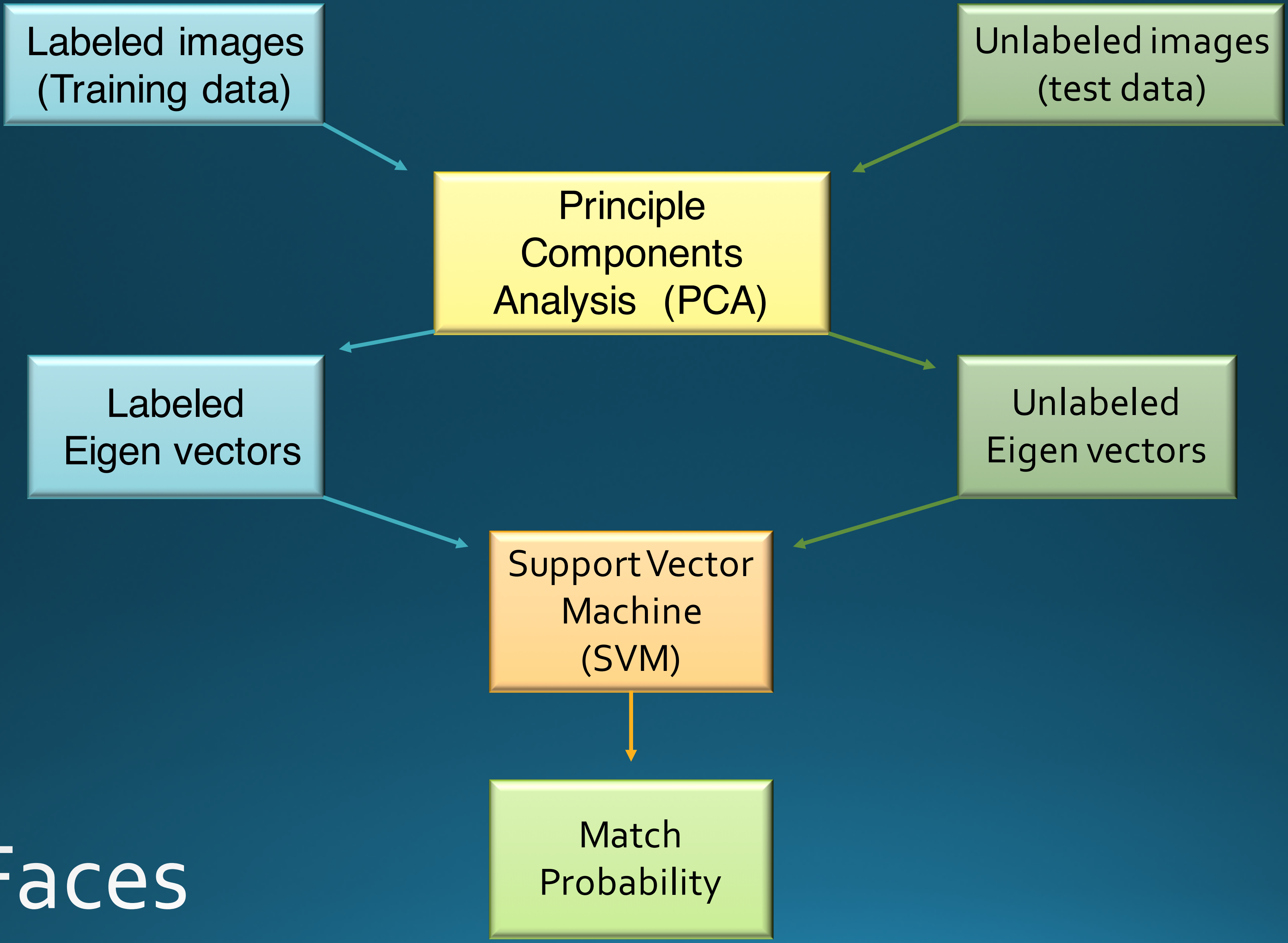
- Seeks to find linear combination of features that characterizes or separates two or more classes of objects or events
- Assumes independent variables are normally distributed
- Defines separation between classes as the ratio of the variance between the classes to the variance within the classes:

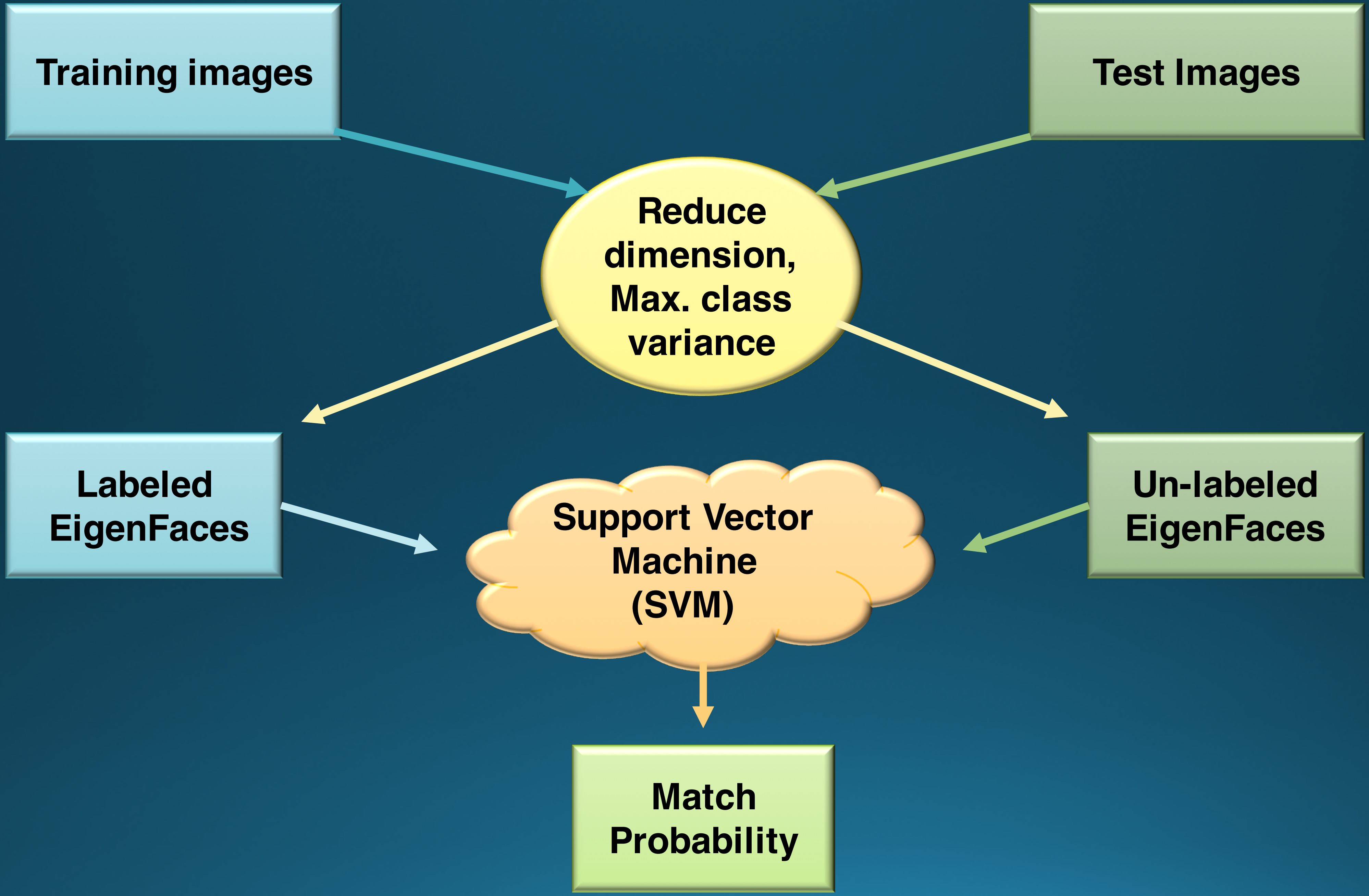


Fisher Linear Discriminant


Labeled
Training images
Unlabeled test
images
Reduce Dimensionality
Maximize variance
Maximize
Clustering
Labeled
Discriminant
Unlabeled
discriminant
SVM
Match
Probability

EigenFaces


Labeled images
(Training data)
Unlabeled images
(test data)
Principle
Components
Analysis (PCA)
Labeled
Eigen vectors
Unlabeled
Eigen vectors
Support Vector
Machine
(SVM)
Match
Probability


Training images
Test Images
Reduce dimension, Max. class variance
Labeled EigenFaces
Support Vector Machine (SVM)
Un-labeled EigenFaces
Match Probability