Analizing the weaknesses of eigenfaces

Nathaniel Yee, Eric Miller

I. Abstract

Overall.

training on a single image of each person leads to not calculating the variance between each person = better results yay!

II. What are eigenfaces?

EIGENFACES (algorithm) was originally developed in 1991 by Matthew A. Turk and Alex P. It is recognized as the original, effective, near-real-time face tracking and recognition algorithm. In the context of this paper, we will examine the recognition component. In particular, we will look at how the size/type of training set affects the accuracy of recognition.

Within the algorithm, the **eigenfaces** are a set of orthonormal vectors in N-dimensional space (where N = number of pixels in each training image). This set of eigenfaces correspond to the principal components of the pixel-pixel covariance matrix constructed from the image training data. As a result, the eigenfaces capture the direction and scale of maximum variance throughout the data. Their corresponding **eigenvalues** represent the amount which the data varies along those axes.

We consider a **compressed image** to be the first k components of the projection of an image vector into ~~the coordinate~~ frame formed by the eigenfaces. Compressing an image simply consists of finding the projection of the image onto the first **k** eigenfaces, and decompressing it requires summing the eigenfaces with coefficients determined by the elements of the compressed form.

To perform facial recognition, we calculate the closest compressed training image (by euclidean distance) to the compressed test image.

math walkthrough of EF.

III. Under what conditions does eigenfaces lose effectivness

The effectiveness of eigenfaces largely depends on the size/type of the training data. Since principal component analysis (PCA) attempts to find the maximum variation between all training images, if we include multiple images of the same person, PCA will attempt to find variation between images of the same person. In other words, we propose that training on multiple photos of the same person will lead to worse recognition and larger clustering.

IV. Figures of Merit

In order to understand the performance of the various incarnations of Eigenfaces, several quantitative and qualitative measures are useful.

Fig. 1: The image representation of an eigenface.

Caption.

A. Accuracy

The first and most obvious measure of algorithmic efficacy is the accuracy algorithm, defined as

$$Acc = \frac{\text{\# of successful recognitions}}{\text{\# of attempted recognitions}}$$

where success is defined as the algorithm's closest chosen closest match being an image of the same person as the given test image. Because both the test and training data are extracted from the same raw dataset, labelling that dataset at the beginning and maintaining those labels throughout the computation makes this analysis trivial.

Unfortunately, because Acc is always an integer multiple of $1/|test\ data|$, it often does not provide enough granularity to distinguish small changes in algorithmic effectiveness.

B. Recognition confidence

To create a more finely-discerning version of Acc, we can analyze the confidence of the algorithm in selecting the match it did for compressed test image t. This involves looking at two compressed images from the training set: c, the nearest training image of the correct person, and w, the nearest training image *not* of a correct ~~wrong~~ person. Because Eigenfaces chooses the closest training face in k-space, if the distance from the test point to c is less than the distance to w ($|t-w| > |t-c|$), then the

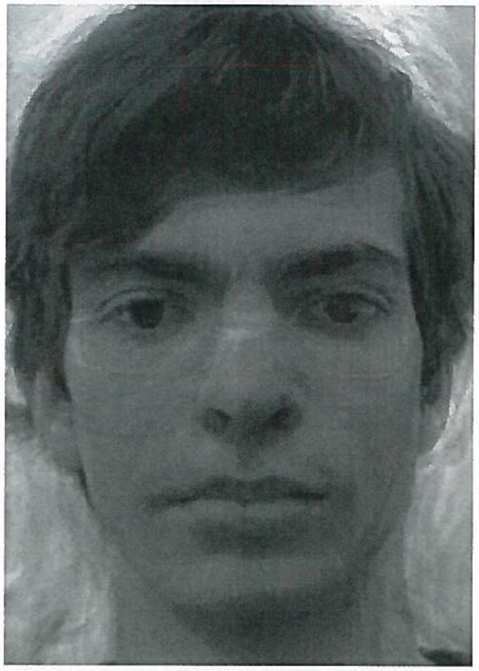


Fig. 2: An image of Eric that has been expressed as linear combinations of the 30 most influential eigenfaces. Since our training set contains people wearing glasses, we see glasses frames in the projected image of Eric.

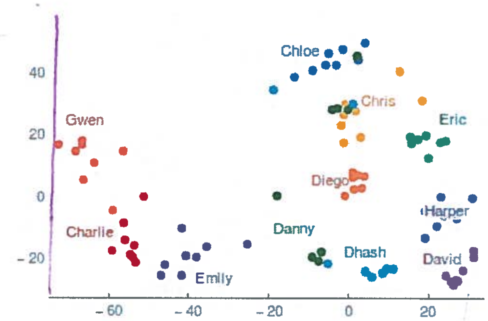


Fig. 3: Shows a graphical representation of people projected onto the second and third eigenvector. Each point represents an image of a person. Each color represents a person. Notice that each person's images cluster together. If we were to perform facial recognition using these eigenvectors, we would get decent results due to the small euclidean distances between images of the same person

algorithm will return the correct match. From this insight, we can compute the Recognition Confidence

$$RC = \frac{1}{|T|} \sum_{t \in T} |t - w| - |t - c|$$

where T is a set of test images.

Notice that each term of this average is positive if eigenfaces would return a correct match for that test image, and negative if it would fail. Thus, larger values of RC correspond to the algorithm being more likely to choose the correct match.

This provides better granularity than the raw Acc measurement, but loses the direct connection to the performance of the algorithm.

C. Clustering Coefficient

At its core, Eigenfaces is an algorithm for projecting the images onto k-dimensional compressed faces such that the compressed faces of the same person are clustered together. To directly analyze the effectiveness of this clustering, we can compute for each person in the dataset the shortest tour distance between the k-dimensional vectors corresponding to each of their images, then average those distances to create a **clustering coefficient** (CC). Intuitively, this represents the size of each cluster, so smaller values are better. Importantly, values of the clustering coefficient are only comparable if the same number of images per person are used, so for all values of the clustering coefficient used in this paper, it was computed from the full 8 faces per person.

V. Performance of two training sets

Next we test eigenfaces for two groups of training and testing data. First, we will train on 43 images. One image per person. Second, we will train on 301 images. Seven images per person.

A. Accuracy

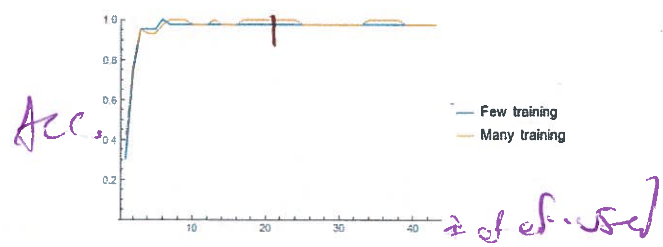


Fig. 4: Shows the accuracy of both algorithms. The Y axis shows accuracy, the X axis shows the number of eigenfaces used. Both algorithms are testing on happy faces (similar to neutral training data) and include neutral faces in training data.

As seen in fig 5, having multiple images of each person does not improve accuracy. In this case, both training sets lead to near 100% accuracy when we train on the similarly looking neutral faces.

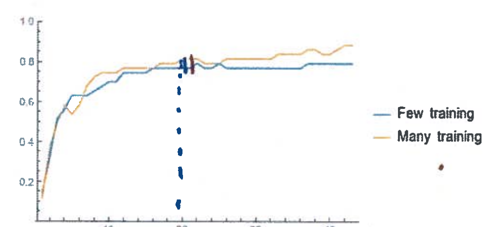


Fig. 5: Shows the accuracy of both algorithms. The Y axis shows accuracy, the X axis shows the number of eigenfaces used. Both algorithms are testing on disgusted faces (not similar to neutral training data) and include neutral faces in training data.

However, as seen in fig **??**, when we test on a more varied expression, training on many faces improves accuracy at higher numbers of eigenfaces.

B. Recognition confidence

C. Clustering Coefficient

Next we will compare the clustering coefficients.

Table 1: Clustering Coefficients

Training image total	Training images per person	Clustering coefficient
41	1	173
301	7	201

As shown in table 1, if we increase the training images per person from one to seven, the clustering coefficient increases from 173 to 201. This confirms part of our initial assumption; as we increase the number of images per person, eigenfaces captures more variation between images of the same person and ultimately creates larger clusters.

VI. Overarching Conclusions / Insights

VII. Appendix: 3D cluster visualizations

VIII. Appendix: Choosing k (number of eigenfaces)

FLOATING flat at rest along both axes, ...

Eric Miller & Nathaniel Yee

1. Abstract
2. ~~Executive Summary~~/Introduction
3. History / What is eigenfaces
4. Explaining eigenfaces/concepts to a lin student
    - a. **Figure: A few Eigenfaces**
    - b. **Equations: up-converting eigenvectors to save compute**
5. Figures of Merit explanation
    - a. Accuracy %
        - i. 1st choice
        - ii. Average number of people till correct guess
        - iii. **Figure: Example recognition grid + FOM**
    - b. Clustering coefficient
        - i. **Equation:**
    - c. Recognition confidence
        - i. **Equation:**
6. Advantages/disadvantages of more samples per person
    - a. Facial recognition accuracy
    - b. Increase cluster size
        - i. **Figures**
    - c. Numerical coefficient
    - d. Explorations with very few people
7. Conclusion
    - a. More samples per person = better / worse in particular ways
    - b. Depends on the number of people
8. Appendix: 3D clustering WHOA
    - a. **A picture of 3D clustering**
9. Appendix: Choosing k (number of eigenfaces)
    - a. **Figure: Graph of recognition accuracy / k**

Analizing the weaknesses of eigenfaces

Nathaniel Yee, Eric Miller

→ includes results

I. ABSTRACT

Training on a single image of each person leads to not calculating the variance between each person = better results yay!

I.5 θ Introduction

II. WHAT ARE EIGENFACES?

EIGENFACES (algorithm) was originally developed in 1991 by Matthew A. Turk and Alex P. It is recognized as the original, effective, near-real-time face tracking and recognition algorithm. cite? In the context of this paper, we will examine the recognition component. In particular, we will look at how the size/type of training set affects the accuracy of recognition.

hard introduce?

Within the algorithm, the **eigenfaces** are a set of orthonormal vectors in N-dimensional space (where N = number of pixels in each training image). This set of eigenfaces correspond to the principal components of the pixel-pixel covariance matrix constructed from the image training data. As a result, the eigenfaces capture the direction and scale of maximum variance throughout the training data. Their corresponding **eigenvalues** represent the amount which the data varies along those axes.

We consider a **compressed image** to be the most significant first k components of the projection of an image vector into the coordinate frame formed by the eigenfaces. Compressing an image involves ~~simply consists of~~ finding the projection of the image onto the first **k** eigenfaces, and decompressing it requires summing the eigenfaces with coefficients determined by the elements of the compressed form.

w/ highest EV-1

To perform facial recognition, we calculate the closest compressed training image (by euclidean distance) to the compressed test image. which?...

III. UNDER WHAT CONDITIONS DOES EIGENFACES LOSE EFFECTIVNESS

? K introduction

The effectiveness of eigenfaces largely depends on the size/type of the training data. Since principal component analysis (PCA) attempts to find the maximum variation between all training images, if we include multiple images of the same person, PCA will attempt to find variation between images of the same person. In other words, we propose that training on multiple photos of the same person will lead to worse recognition and larger clustering.

IV. FIGURES OF MERIT

In order to understand the performance of the various incarnations of Eigenfaces, several quantitative and qualitative measures are useful.

really? much?

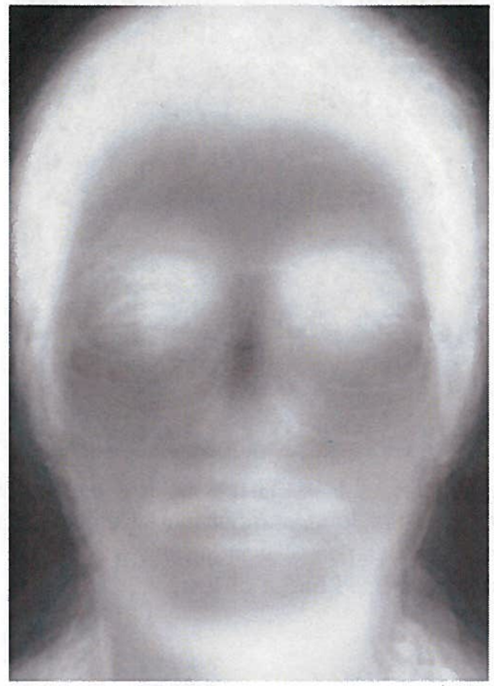


Fig. 1: The image representation of an eigenface.

neutral dataset

A. Accuracy

The first and most obvious measure of algorithmic efficacy is the accuracy algorithm, defined as

$$Acc = \frac{\text{\# of successful recognitions}}{\text{\# of attempted recognitions}}$$

where success is defined as the algorithm's closest chosen closest match being an image of the same person as the given test image. Because both the test and training data are extracted from the same raw dataset, labelling that dataset at the beginning and maintaining those labels throughout the computation makes this analysis trivial.

Unfortunately, because Acc is always an integer multiple of $1/|test\ data|$, it often does not provide enough granularity to distinguish small changes in algorithmic effectiveness.

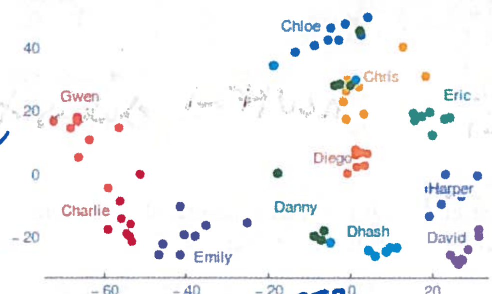
B. Recognition confidence

To create a more finely-discerning version of Acc, we can analyze the confidence of the algorithm in selecting the match it did for compressed test image t. This involves looking at two compressed images from the training set: c, the nearest training image of the correct person, and w, the nearest training image *not* of a correct ~~wrong~~ person. Because Eigenfaces chooses the closest training face in k-space, if the distance from the test point to c is less than the distance to w ($|t-w| > |t-c|$), then the

Fig. 2: An image of Eric that has been expressed as linear combinations of the 30 most influential eigenfaces. Since our training set contains people wearing glasses, we see glasses frames in the projected image of Eric.

ei Label eigenfaces

EF 2



EF3

Fig. 3: Shows a graphical representation of people projected onto the second and third eigenvector. Each point represents an image of a person. Each color represents a person. Notice that each person's images cluster together. If we were to perform facial recognition using these eigenvectors, we would get decent results due to the small euclidean distances between images of the same person

algorithm will return the correct match. From this insight, we can compute the Recognition Confidence

$$RC = \frac{1}{|T|}\sum_{t\in T}|t-w| - |t-c|$$

where T is a set of test images.

Notice that each term of this average is positive if eigenfaces would return a correct match for that test image, and negative if it would fail. Thus, larger values of RC correspond to the algorithm being more likely to choose the correct match.

This provides better granularity than the raw Acc measurement, but loses the direct connection to the performance of the algorithm.

C. Clustering Coefficient

At its core, Eigenfaces is an algorithm for projecting the images onto k-dimensional compressed faces such that the compressed faces of the same person are clustered together. To directly analyze the effectiveness of this clustering, we can compute for each person in the dataset the shortest tour distance between the k-dimensional vectors corresponding to each of their images, then average those distances to create a **clustering coefficient** (CC). Intuitively, this represents the size of each cluster, so smaller values are better. Importantly, values of the clustering coefficient are only comparable if the same number of images per person are used, so for all values of the clustering coefficient used in this paper, it was computed from the full 8 faces per person.

Training Testing

V. Performance of Two Training Sets

Next we test eigenfaces for two groups of training and testing data. First, we will train on 43 images. One image per person. Second, we will train on 301 images. Seven images per person.

A. Accuracy

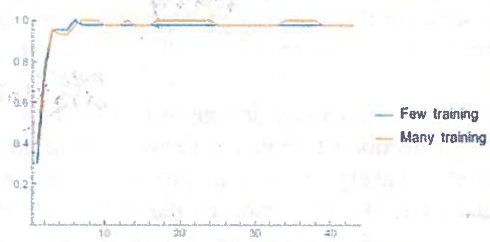


Fig. 4: Shows the accuracy of both algorithms. The Y axis shows accuracy, the X axis shows the number of eigenfaces used. Both algorithms are testing on happy faces (similar to neutral training data) and include neutral faces in training data.

As seen in fig 5, having multiple images of each person does not improve accuracy. In this case, both training sets lead to near 100% accuracy when we train on the similarly looking neutral faces.

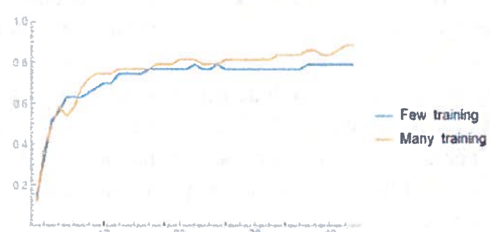


Fig. 5: Shows the accuracy of both algorithms. The Y axis shows accuracy, the X axis shows the number of eigenfaces used. Both algorithms are testing on disgusted faces (not similar to neutral training data) and include neutral faces in training data.

However, as seen in fig **??**, when we test on a more varied expression, training on many faces improves accuracy at higher numbers of eigenfaces.

B. Recognition confidence

C. Clustering Coefficient

Next we will compare the clustering coefficients.

Table 1: Clustering Coefficients		
Training image total	Training images per person	Clustering coefficient
41	1	173
301	7	201

As shown in table 1, if we increase the training images per person from one to seven, the clustering coefficient increases from 173 to 201. This confirms part of our initial assumption; as we increase the number of images per person, eigenfaces captures more variation between images of the same person and ultimately creates larger clusters.

VI. Overarching Conclusions / Insights

VII. Appendix: 3D cluster visualizations

VIII. Appendix: Choosing k (number of eigenfaces)

FLOATING flat at rest along both axes, ...

Eric Miller & Nathaniel Yee

- Methodology
- Goals
- What's this paper abou

Describe process

1. Abstract
2. ~~Executive Summary~~/Introduction
3. History / What is eigenfaces
4. Explaining eigenfaces/concepts to a lin student — Eric
    a. **Figure: A few Eigenfaces**
    b. **Equations: up-converting eigenvectors to save compute**
5. Figures of Merit explanation — tuning
    a. Accuracy %
        i. 1st choice
        ii. Average number of people till correct guess
        iii. **Figure: Example recognition grid + FOM**
    b. Clustering coefficient
        i. **Equation:**
    c. Recognition confidence
        i. **Equation:**
6. Advantages/disadvantages of more samples per person — Meet
    a. Facial recognition accuracy
    b. Increase cluster size
        i. **Figures**
    c. Numerical coefficient
    d. Explorations with very few people

how choose

Defining two algorithms

7. Conclusion
    a. More samples per person = better / worse in particular ways
    b. Depends on the number of people
8. Appendix: 3D clustering WHOA
    a. **A picture of 3D clustering**
9. Appendix: Choosing k (number of eigenfaces)
    a. **Figure: Graph of recognition accuracy / k**

Eric