

A Study On The Efficacy of Using Eigenfaces In Image Classification

Jacob T. Cassady*, Dimitri Medina†

Johns Hopkins University, Whiting School of Engineering, Baltimore, MD, 21206

The increased use of image classification in technology today has incited a rise in research and development for new approaches in facial detection and identification models. Two common problems in image classification are storing large datasets and model training costs. One approach to achieving dimensionality reduction while maintaining performance is Principal Component Analysis where a subset of eigenvectors, also known in the domain of facial detection as “eigenfaces”, are used to represent the data in a lower dimensionality space. This paper presents an image classification model based on eigenfaces and support vector machines using the Amsterdam Dynamic Facial Expression Set (ADFES) dataset. Implementation of an image classification model is described, and performance analysis of the model is presented with a focus on the efficacy of using eigenfaces when training the model.

I. Nomenclature

<i>ADFES</i>	=	Amsterdam Dynamic Facial Expression Set
<i>AICE</i>	=	Amsterdam Interdisciplinary Centre for Emotion
<i>PCA</i>	=	Principal Component Analysis
<i>SVM</i>	=	Support Vector Machine

II. Introduction

Image classification has become an innovation of interest in recent years due to the parallel advances in machine learning techniques, digital camera technology, and other similar computational fields of science. From security monitoring to access control, image classification has become an important tool in many diverse disciplines. Two common problems in image classification are storing large datasets and model training costs.

One approach to achieving dimensionality reduction while maintaining a representation of the data is Principal Component Analysis (PCA) [1]. In the field of facial recognition, PCA can remove the reliance on the isolation of features such as eyes and ears, and instead takes a mathematical approach to defining and characterizing face images.

*Graduate Student, Robotics and Autonomous Systems

†Graduate Student, Mechanical Engineering

PCA uses a subset of eigenvectors, also known in facial recognition as “eigenfaces”, to represent the data in a lower dimensionality space. If a certain characteristic is prominent in a collection of images, the eigenface corresponding to the characteristic has a larger eigenvalue and represents a higher explained variance of the dataset [2]. After utilizing PCA, eigenvectors of lesser importance can be removed to reach a desired level of dimensionality reduction with the trade-off of reduced explained variance of the initial dataset.

SVMs are supervised learning algorithms designed to find the optimal hyperplane that maximizes the margin between the closest data points of opposite classes [3]. SVMs suffer high computational costs when the number of features is large. Using eigenvectors calculated with PCA has been shown to work well with Support Vector Machines (SVMs) for classification in a variety of domains [4][5][6].

This paper focuses on analyzing the efficacy of using eigenfaces when performing image classification of the Amsterdam Dynamic Facial Expression Set (ADFES) dataset using SVMs. Section III provides a description of the ADFES dataset. Section IV introduces the model used for image classification including the mathematics of eigenfaces and SVMs. Section V describes the implementation of a classification model using eigenfaces and SVMs with the Python programming language. Section VI presents analysis of the classification model including the efficacy of using eigenfaces in image classification. This paper concludes with a discussion of the results and future work in sections VII and VIII respectively.

III. Amsterdam Dynamic Facial Expression Set

The ADFES dataset was developed by the University of Amsterdam’s Amsterdam Interdisciplinary Centre for Emotion (AICE) [7]. The ADFES dataset includes expressions displayed by 22 models from Northern-European and Mediterranean locations. There are ten emotions included in ADFES dataset: anger, contempt, disgust, embarrassment, fear, joy, neutral, pride, sadness, and surprise. The ADFES dataset includes both videos and still images. Each media is labeled with a gender: male or female. This paper will utilize the 217 still images from the ADFES dataset only. Figure 1 includes an example of a still image from the ADFES dataset. Table 1 shows the number of classes per target in the dataset and the images per class. Each image has a width of 720, a height of 576, and three 8-bit color channels: red, green, and blue.

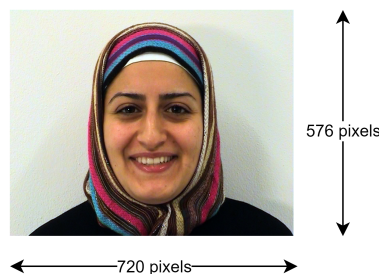


Fig. 1 ADFES Example Image with targets: Mediterranean, Female and Joy

Target	Class Count	Images Per Class
Geographic Tag	2	{Northern European: 120, Mediterranean: 97}
Gender Tag	2	{Male: ≈ 120 , Female: ≈ 100 }
Model Identification	22	≈ 10
Emotion	10	≈ 22

Table 1 Target Class Distributions

IV. Model

Image classification will be accomplished using eigenfaces and SVMs. Calculation of eigenfaces, or more generally eigenvectors, will be calculated using Principal Component Analysis (PCA). Section IV.A describes the shapes of the feature and target matrices before the eigenface dimensionality reduction described in Section IV.B. Section IV.C describes the mathematics of a SVM.

A. Features and Targets

Each image I of the dataset follows the representation shown in equation 1. As previously mentioned, each image begins with shape (576 *height*, 720 *width*, 3 *color channels*).

$$I(i) = \begin{bmatrix} (r_{(0,0)}, g_{(0,0)}, b_{(0,0)}) & (r_{(0,1)}, g_{(0,1)}, b_{(0,1)}) & \dots & (r_{(0,719)}, g_{(0,719)}, b_{(0,719)}) \\ (r_{(1,0)}, g_{(1,0)}, b_{(1,0)}) & (r_{(1,1)}, g_{(1,1)}, b_{(1,1)}) & \dots & (r_{(1,719)}, g_{(1,719)}, b_{(1,719)}) \\ \dots & \dots & \dots & \dots \\ (r_{(575,0)}, g_{(575,0)}, b_{(575,0)}) & (r_{(575,1)}, g_{(575,1)}, b_{(575,0)}) & \dots & (r_{(575,719)}, g_{(575,719)}, b_{(575,719)}) \end{bmatrix} \quad (1)$$

The images are first flattened as shown in equation 2.

$$I(i)_{flat} = \begin{bmatrix} r_{(0,0)} & g_{(0,0)} & b_{(0,0)} & \dots & r_{(0,719)} & g_{(0,719)} & b_{(0,719)} & \dots & r_{(575,719)} & g_{(575,719)} & b_{(575,719)} \end{bmatrix} \quad (2)$$

The flattened images are then stacked on top of each other to create a matrix of features as shown in equation 3.

$$features = \begin{bmatrix} I(0)_{flat} \\ I(1)_{flat} \\ \dots \\ I(216)_{flat} \end{bmatrix}, targets = \begin{bmatrix} T(0) \\ T(1) \\ \dots \\ T(216) \end{bmatrix} \quad (3)$$

The final feature matrix will be of shape (217, 1,244,160). The target matrix is created by retrieving the column matching the target data. If the target data is in the form of a label, each string is encoded into an integer matching the class. In the example of gender, males would be encoded to 1 and female encoded to 2. The final target matrix will be of shape (217, 1).

Before feeding into the model for training, the data was randomly shuffled and then split into training and validation sets. With an 80-20 training and validation split, 172 images and associated targets were placed in the training set and 44 images and associated targets were placed in test set. The training features and validation features were of shapes (172, 1,244,160) and (44, 1,244,160) respectively. The training targets and validation targets were of shapes (172, 1) and (44, 1) respectively.

B. Principal Component Analysis

PCA can be used to achieve dimensionality reduction while maintaining a representation of the data [1]. PCA uses a subset of eigenvectors, also known in facial recognition as “eigenfaces”, to represent the data in a lower dimensionality space. Eigenvectors will each have an associated eigenvalue which is a scalar and is a measure of the eigenvector prominence in the dataset.

To calculate the eigenfaces, processing of the face images and the calculation of the covariance matrix must be done. Each face image $I(i)$ will be represented as a vector Γ_n . The vectors Γ_n will then be used to calculate the average matrix Ψ of each component as shown in equation 4.

$$\Psi = \frac{1}{N} \sum_{n=1}^M \Gamma_n \quad (4)$$

The resulting matrix will then be subtracted from each face image and stored in the variable Φ_n as shown in equation 5. Φ_n is used to calculate the covariance matrix C as shown in equation 6.

$$\Phi_n = \Gamma_n - \Psi \quad (5)$$

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T, \text{ where } A = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (6)$$

The covariance matrix has the eigenfaces u_i and their respective eigenvalues v_i as shown in equation 7, where u_i is solved with equation 8. Finally, it is important to normalize u_i such that $\|u_i\| = 1$.

$$A^T A v_i = u_i v_i \quad (7)$$

$$u_i = \sum_{k=1}^M v_{lk} \Phi_k, \text{ where } l = 1, 2, \dots, M \quad (8)$$

The training features and validation features after the eigenface transform were of shapes $(172, E_{count})$ and $(44, E_{count})$ respectively where E_{count} is the number of components (eigenvectors or eigenfaces) from PCA retained. As described in Section VI.B, different values of E_{count} were chosen for different targets. A higher E_{count} value means lower dimensionality reduction and higher explained variance of the input feature space.

C. Support Vector Machine

SVMs are supervised learning algorithms designed to find the optimal hyperplane that maximizes the margin between the closest data points of opposite classes [3]. Multiclass classification can be achieved using a One-to-One or a One-to-Rest approach. The closest datapoints are also called “support vectors”. SVM classifiers are based on the class of hyperplanes shown in equation 9.

$$(w \bullet x) + b = 0, w \in \mathbb{R}^N, b \in \mathbb{R} \quad (9)$$

The optimal hyperplane can be found by solving a constrained optimization problem whose solution w has an expansion shown in equation 10 where x_i is one training example. Solutions to SVMs can be generated using quadratic programming.

$$w = \sum_{i=1}^N v_i x_i \quad (10)$$

SVMs perform a nonlinear separation from the input space into a higher-dimensional “feature space” F as shown in equation 11. The mapping performed by the SVM is called the kernel function $K(x, y)$. This paper will focus on the linear kernel defined by equation 12

$$\Phi : \mathbb{R}^N \rightarrow F \quad (11)$$

$$K(x, y) = \Phi(x) \bullet \Phi(y) \quad (12)$$

The decision function for the SVM is shown in equation 13 where v_i are the parameters calculated using quadratic programming.

$$f(x) = \text{sign} \left(\sum_{i=1}^l v_i \bullet k(x, x_i) + b \right) \quad (13)$$

V. Implementation

To analyze the efficacy of eigenfaces for maintaining model performance while reducing the dimensionality of the data, the model described in section IV was implemented in Python 3.8. Table 3 lists the 3rd party libraries used in the implementation. Section V.A describes the data preprocessing steps taken before feeding the data into the model. Section V.B describes the implementation of the model.

A. Data Preprocessing

Images downloaded from the ADFES dataset were placed in two directories named after the geographic tag. Files from the ADFES dataset included the regional model identification, emotion, and gender tag in the file name. A global model identification value was calculated using equation 14 where ID_{Max} is the largest regional model id in the dataset, G is an enumeration dependent on the geographic tag on the image, and S is an enumeration dependent on the gender tag on the image. A python script was written to place image data into a pandas [8] DataFrame with rows representing an image with associated metadata and columns representing the geographic tag, model identification, emotion, gender tag, and pixel values. After preprocessing the data, the pandas data structure was stored using a Python pickle file.

$$ModelID_{Global} = ModelID_{Regional} + (ID_{Max} + 1)(G + 2S) \quad (14)$$

To format the data for usage by the model, pixel data was flattened for each image into a single row as shown in Equation 2. Each row of image data was then stacked on top of each other to form a matrix of features of shape (217, 1,244,160). The training column was extracted from the pandas DataFrame and formatted as a NumPy array of shape (217,1). The sci-kit learn [9] LabelEncoder preprocessing class was used to convert string labels into integer values. Data was then randomly shuffled and split into train and validation sets using sci-kit learn's train_test_split method.

B. Model Implementation

To calculate the eigenvalues of the data, sci-kit learn's Principal Component Analys (PCA) class was used. The singular value decomposition solver was set to "randomized" - meaning it calculated the eigenfaces using the method described by Halko et al. [10]. The sci-kit learn SVC class was used for support vector machine classification. The regularization parameter $C = 3$ and a linear kernel were used.

VI. Analysis

The primary benefit of using eigenfaces is dimensionality reduction, therefore the analysis section focuses on the trade off between using eigenfaces to reduce datasize and maintaining model performance. Section VI.A provides analysis of Eigenface Count's (E_{count}) effect on model training performance. Section VI.B presents the performance of the best model for each target in the ADFES dataset with the largest amount of dimensionality reduction.

A. Eigenface Analysis

As previously stated, the number of eigenfaces retained from PCA determines the amount of explained variance represented by the eigenface features when compared to the input features. Figure 2 shows the performance of the SVM on each target with E_{count} values ranging from [2, 10]. This analysis was performed 5 times for each data point. The left Y-axis represents the accuracies of the models. The center blue dots represent the average model accuracy over the five tests while the blue lines represent the standard deviations of the accuracies. The purple lines represent the amount of explained variance in the eigenface features used in training the model. Figure 5 shows the performance of the SVM on each target with E_{count} values ranging from [10, 125].

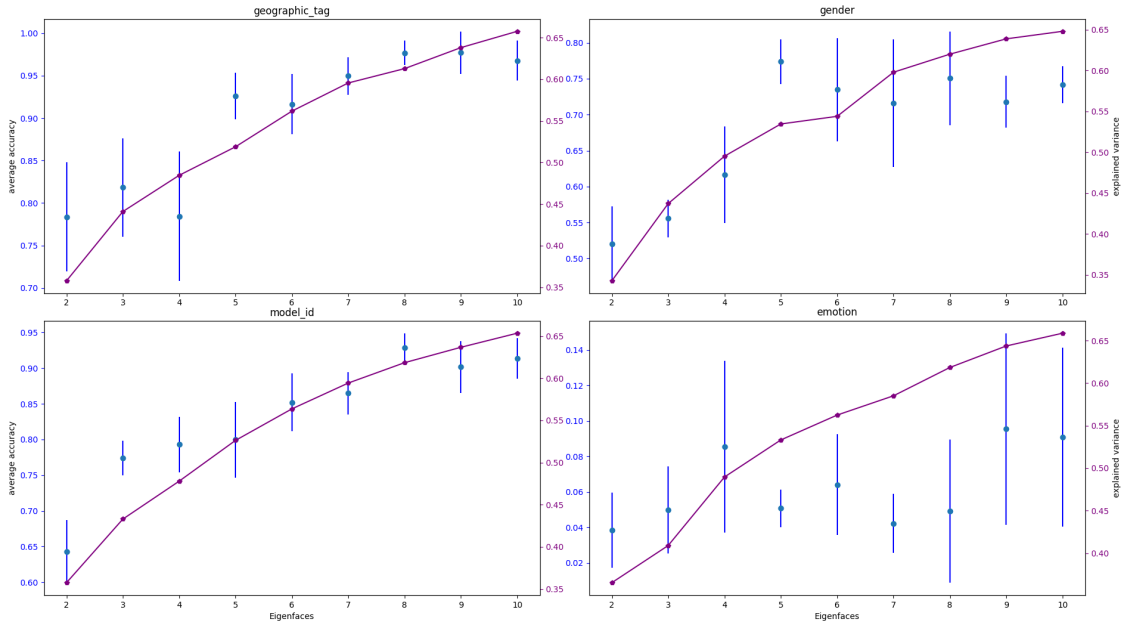


Fig. 2 Eigenface Efficacy Analysis (2-10 eigenfaces). The x-axis of each plot is the number of eigenfaces used in the training. The left y-axis for each plot is the average weighted accuracy over 5 tests. The right y-axis for each plot is the total explained variance of the eigenfaces used in the training.

Analyzing the eigenfaces directly can provide valuable insight into the dataset. As shown in Figure 2, the top 9 eigenfaces shown in Figure 3 represent more than 65% of the explained variance in the input features.

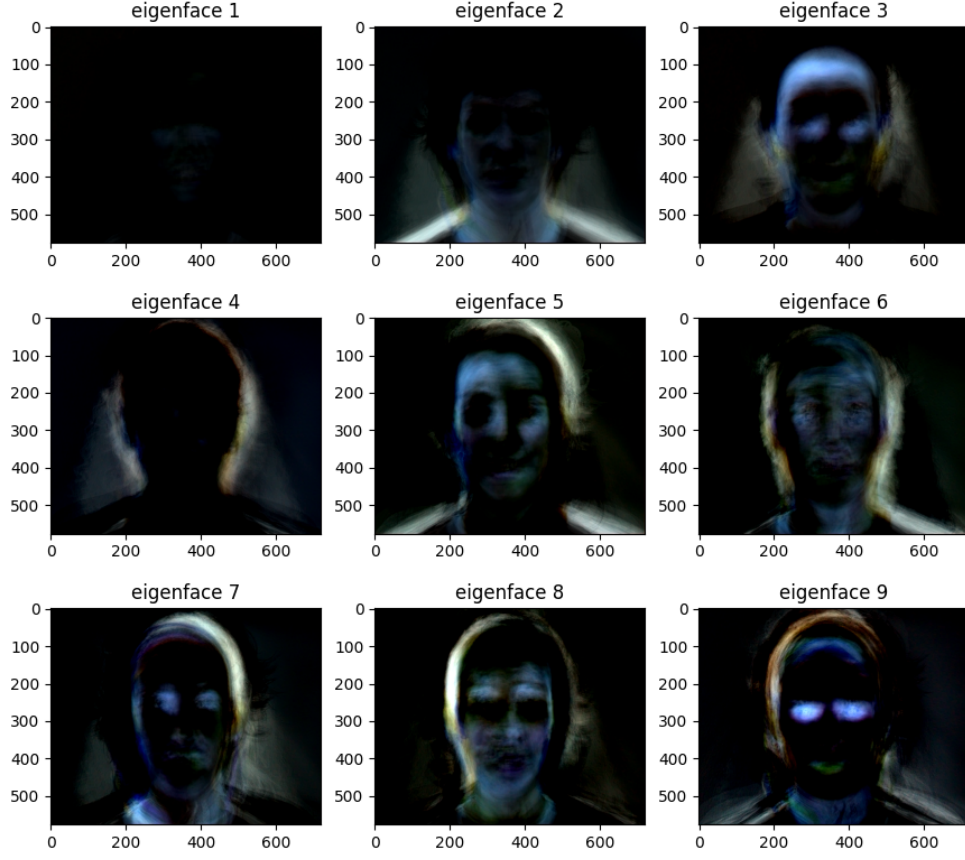


Fig. 3 Top 9 Eigenfaces

B. Classification Analysis

Before performing PCA, features and targets were randomly shuffled. 80% of the data or 172 images and associated targets were placed in training set. 20% of the data or 44 images and associated targets were placed in test set. Models were trained to classify each target in the dataset: geographic tag, gender tag, model identification, and emotion. Each model was trained for a maximum of 5 times and the best outcome was retained. The F_1 score, also known as the harmonic mean of the precision and recall, is the performance metric chosen for comparing models. Equation 15 for F_1 score can be found below.

$$F_1 = 2 \frac{\text{precision} \bullet \text{recall}}{\text{precision} + \text{recall}} \quad (15)$$

Table 2 presents the performance of the model on each target. Eigenface count (E_{count}) represents the number of eigenfaces with the highest explained variances retained after performing PCA. The explained variance column represents the sum of explained variances of all eigenfaces used in the target's model. E_{count} values were chosen analytically from inspecting Figure 5 and selecting the smallest E_{count} with the best performance. Data size reduction

was calculated with the formula shown in equation 16.

$$\text{Data Size Reduction} = \frac{\text{size}(\text{EigenfaceFeatures}) - \text{size}(\text{Features})}{\text{size}(\text{Features})} \bullet 100\% \quad (16)$$

Target	E_{count}	Explained Variance	Data Size Reduction	Best F_1 Score
Geographic Tag	50	$\approx 87\%$	-99.9960%	1.00
Gender Tag	65	$\approx 90\%$	-99.9948%	1.00
Model Identification	50	$\approx 87\%$	-99.9960%	1.00
Emotion	75	$\approx 92\%$	-99.9940%	0.293

Table 2 Best Model Performances

Confusion matrices were also generated to provide insight into the model's performance on the validation set per class of the target. The confusion matrices for geographic tag, gender, model identification, and emotion can be found in Figures 6, 7, 8, and 4 respectively.

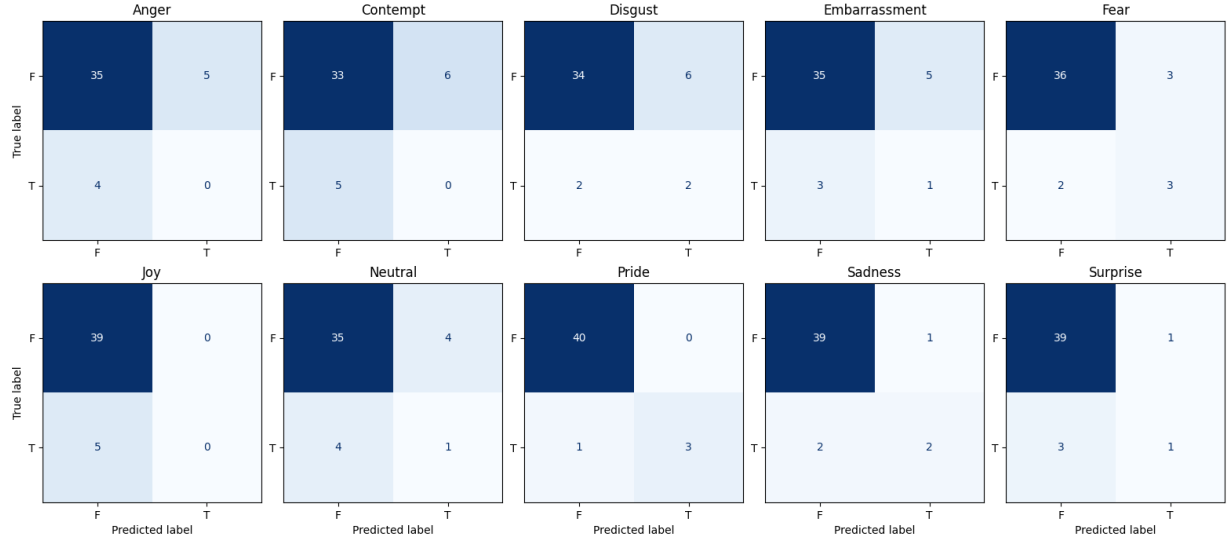


Fig. 4 Emotion Confusion Matrices

VII. Discussion

Eigenfaces retain high classification accuracy while achieving dimensionality reduction. Figure 2 shows the performance of the model on each target with 2-10 eigenfaces and Figure 5 shows the performance of the model on each target with 10-125 eigenfaces. With only the top nine eigenfaces, shown in Figure 3, the model was able to achieve an

F_1 Score of more than 90% on geographic tag and model identification targets. Table 2 shows a perfect F_1 Score was achieved on 3 of the 4 targets with a data size reduction of more than 99.99% from using eigenfaces.

High classification accuracy was achieved on all targets except emotion. Table 2 shows the best performance of the support vector machine classifier on each target. The confusion matrix for emotion shown in Figure 4 provides insight into the model’s performance issues with emotion classification. The model is over-classifying for anger, contempt, disgust, embarrassment, and neutral while under-classifying for emotions like joy, pride, sadness, and surprise. The poor performance of the model on classifying emotion is possibly due to the relatively low amount of data per emotion class in the ADFES still images.

The ADFES dataset was chosen to test the benefit of eigenfaces because of the low development cost for preprocessing the data. Additionally, the ADFES dataset includes still images that are centered on the model’s face. The consistent positioning provides good insight into what an eigenface, and more generally what an eigenvector, represents in the context of PCA. Because still, well-formatted, images were used to train and validate the model described in this paper is unlikely to perform well in computer vision systems that have dynamic angles on their subject.

VIII. Future Work

The choice to focus on still images was secondary to schedule. The model’s performance on classifying emotions could be greatly improved by increasing the number of images per emotion. Images could be extracted from the video data in the ADFES dataset. Images could also be generated using data augmentation methods [11]. The model would also benefit from diverse camera angles.

The ADFES dataset included images with participants from two geographic tags: Northern European and Mediterranean. It is important to understand the populations represented in a dataset to understand what populations the model has been validated on. For consistent performance across larger populations, there needs to be more diversity in the training and validation sets [12].

Appendix

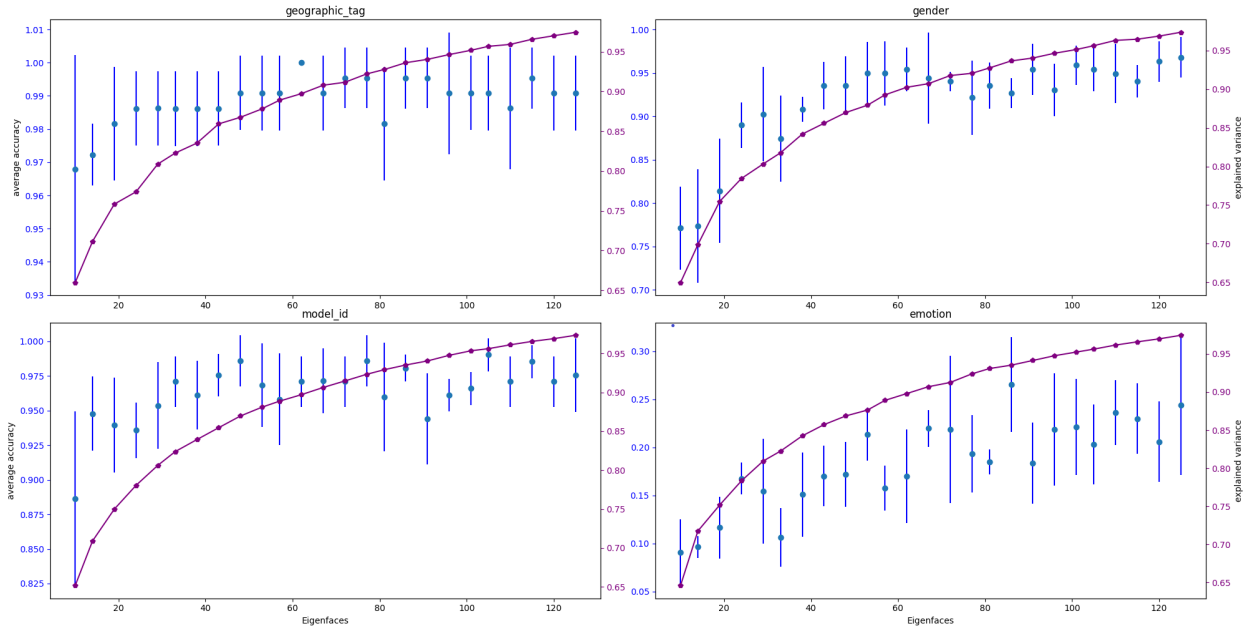


Fig. 5 Eigenface Efficacy Analysis (10-125 eigenfaces). The x-axis of each plot is the number of eigenfaces used in the training. The left y-axis for each plot is the average weighted accuracy over 5 tests. The right y-axis for each plot is the total explained variance of the eigenfaces used in the training.

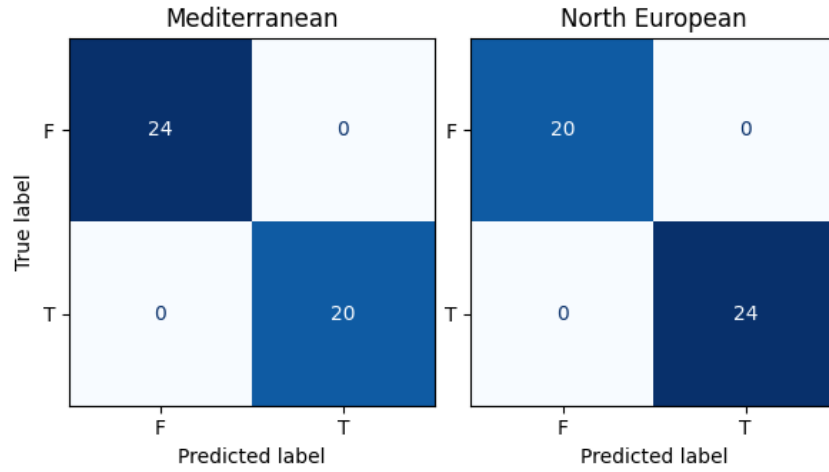


Fig. 6 Geographic Tag Confusion Matrices

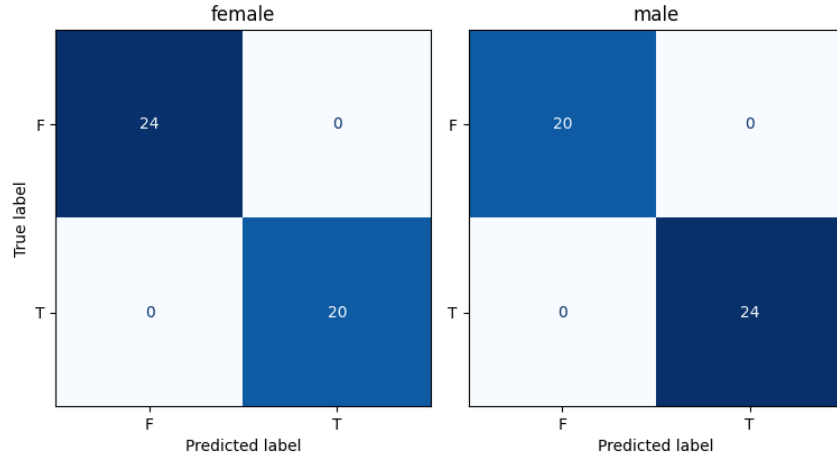


Fig. 7 Gender Tag Confusion Matrices

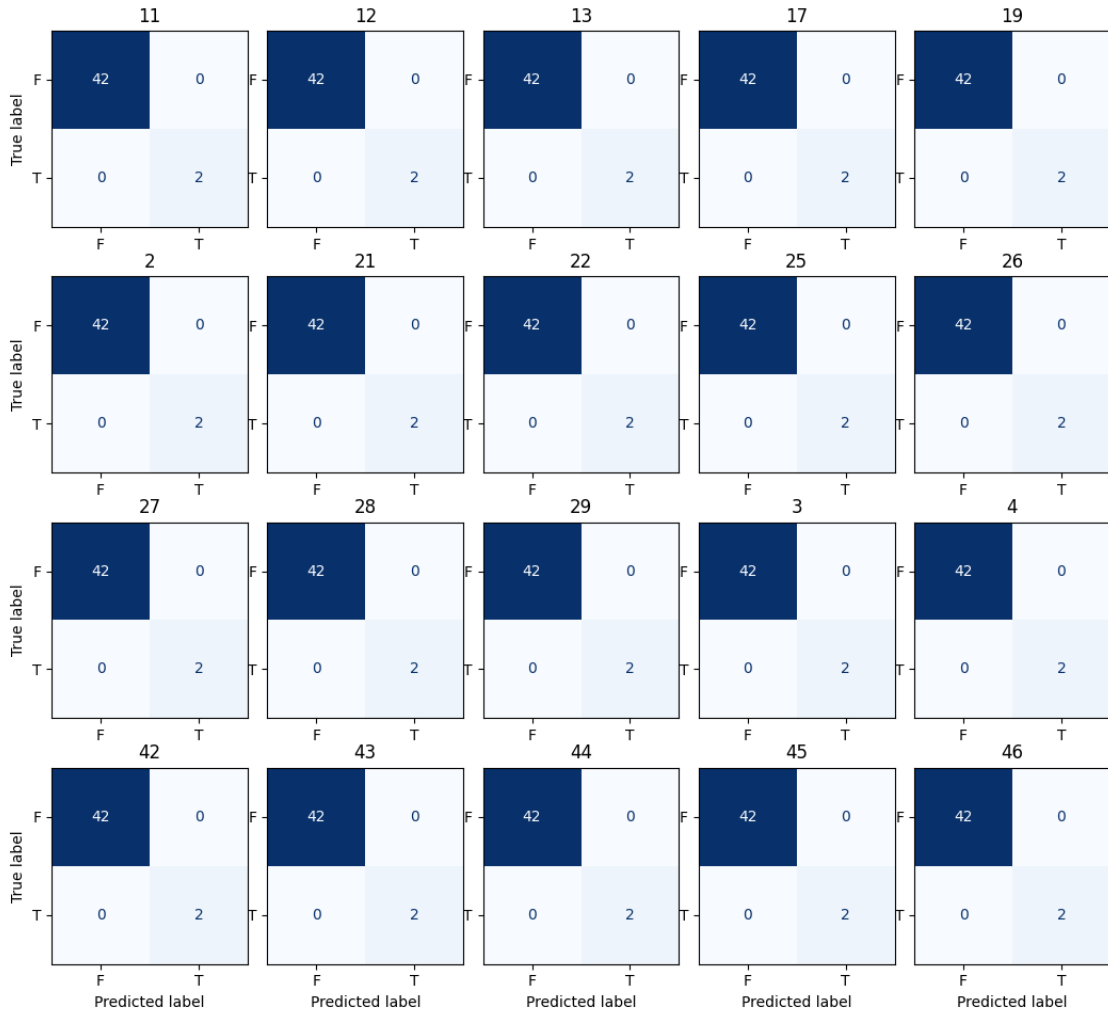


Fig. 8 Model Identification Confusion Matrices. Only 20 of the 22 model identities shown. The remaining 2 confusion matrices are the same as those shown.

Name	Description	Use
NumPy	Comprehensive mathematical functions [13]	Mathematical manipulations of data
pandas	Data analysis and manipulation tool [8]	Store and filter data
matplotlib	Library for creating visualizations [14]	Data inspection and performance plotting
sci-kit learn	Tools for predictive data analysis [9]	PCA, train/test split, and SVM model

Table 3 3rd Party Libraries Used

Acknowledgments

This work was performed as a final paper for the Mathematical Methods for Engineers class at Johns Hopkins University taught by Professor George Nakos.

References

- [1] Abdi, H., and Williams, L. J., "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, Vol. 2, No. 4, 2010, pp. 433–459.
- [2] Turk, M. A., and Pentland, A. P., "Face recognition using eigenfaces," *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, IEEE Computer Society, 1991, pp. 586–587.
- [3] Hearst, M., Dumais, S., Osuna, E., Platt, J., and Scholkopf, B., "Support vector machines," *IEEE Intelligent Systems and their Applications*, Vol. 13, No. 4, 1998, pp. 18–28. <https://doi.org/10.1109/5254.708428>.
- [4] Mangasarian, O. L., and Wild, E. W., "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 28, No. 1, 2005, pp. 69–74.
- [5] Alvarez, I., Górriz, J. M., Ramírez, J., Salas-Gonzalez, D., López, M., Segovia, F., Puntonet, C. G., and Prieto, B., "Alzheimer's diagnosis using eigenbrains and support vector machines," *Bio-Inspired Systems: Computational and Ambient Intelligence: 10th International Work-Conference on Artificial Neural Networks, IWANN 2009, Salamanca, Spain, June 10-12, 2009. Proceedings, Part I 10*, Springer, 2009, pp. 973–980.
- [6] Melišek, J. M., and Pavlovicová, M. O., "Support vector machines, PCA and LDA in face recognition," *J. Electr. Eng.*, Vol. 59, No. 203-209, 2008, p. 1.
- [7] van der Schalk, J., Hawk, S. T., Fischer, A. H., and Doosje, B., "Moving faces, looking places: Validation of the Amsterdam Dynamic Facial Expression Set (ADFES)," *Emotion*, Vol. 11, No. 4, 2011, pp. 907–920. <https://doi.org/10.1037/a0023853>.
- [8] NumFOCUS, Inc., "pandas," 2024. URL <https://pandas.pydata.org/>.

- [9] scikit-learn Development Team, “scikit-learn,” , 2024. URL <https://scikit-learn.org/stable/>.
- [10] Halko, N., Martinsson, P.-G., and Tropp, J. A., “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM review*, Vol. 53, No. 2, 2011, pp. 217–288.
- [11] Maharana, K., Mondal, S., and Nemade, B., “A review: Data pre-processing and data augmentation techniques,” *Global Transitions Proceedings*, Vol. 3, No. 1, 2022, pp. 91–99.
- [12] Zou, J., and Schiebinger, L., “AI can be sexist and racist—it’s time to make it fair,” , 2018.
- [13] NumPy Development Team, “NumPy,” , 2024. URL <https://numpy.org/>.
- [14] Matplotlib Development Team, “matplotlib,” , 2024. URL <https://matplotlib.org/>.