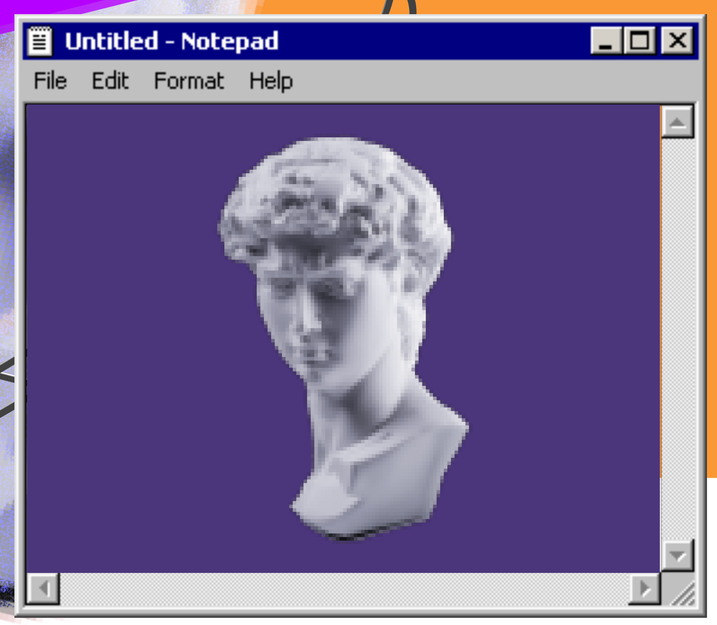
MASK-UNMASK FACE DETECTION

BY GROUP 3

Gu Zhenrong, Lin Pengyun
Su Yuexin, Zhao Songwen



OVERVIEW

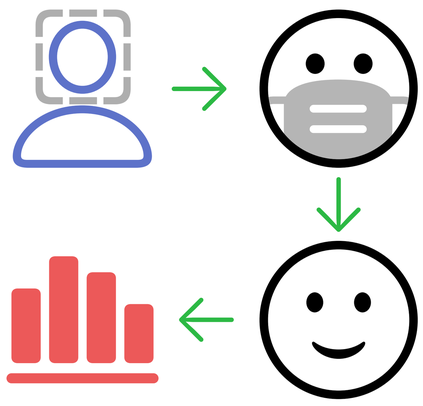
We created a unified model based on Principal Component Analysis and Support Vector Machine for all the classification tasks. To obtain more robustness against obscured faces and extendability, we introduce *Eigenface with Importance Weights* (EIW), and modified the decomposition process of PCA for processing non-orthogonal EIW. As a result, the eigenfaces of the model can try to estimate the face with better approximation in regions with higher weights and try not to estimate the mask, thus predict the covered part of face based on seen data and likelihood.

For preprocessing, we use the dlib module and its pre-trained random forest model to extract the feature points from the image. To make the eigenfaces clearer, we connect the top point of nose and middle point of jaw as the orientation vector and rotate the image so that the vector is vertical. Then, the minimal bounding square of the face couple with some margin is extracted and scaled to 150x150 as face images.

To detect whether the face is wearing a mask, we use a extraction method dedicated in object detection to extract features — *Histogram of Oriented Gradients* (HOG). The features are fed into an SVM to judge whether the face is wearing a mask.

To complete classifing the belonging of the faces, after feature extraction of EIW, the features of an image is processed in another SVM to calculate the possibility of belonging to one person. We calculate its confidence level by maximum probablity to determine whether the face belongs to a person in dataset.

We finally achieved an accuracy of 0.76 in bonus level, recognizing masked faces on our model which is only trained on unmasked dataset. Its accuracy is similar to our model's which trained on both masked and unmasked model.



PREPROCESSING

For preprocessing, we use the *dlib* module and its pre-trained random forest model to extract the feature points from the image. To make the eigenfaces clearer, we connect the top point of nose and middle point of jaw as the orientation vector and rotate the image so that the vector is vertical. Then, the minimal bounding square of the face couple with some margin is extracted and scaled to the same size as face images. We generate the masked database by making a polygon with corespondent feature points.

We keep all the three channels in preprocessing. We find out that keeping RGB values of those images increases the accuracy of detection by 8%.

Fig. 2 *Preprocessing*
From left to right: Original image; feature points and orientation vector; rotated face and mask creation

FEATURE EXTRACTION

We analyze the feature of the face images by conducting Principal Component Analysis.

We directly use NumPy's numpy.svd method to perform PCA. As a result, the correlation matrix is calculated instead of covariance matrix, which brings greater temporal performance and consistency to the model.

According to the composition process $X = U\Sigma V^T$, since U is orthogonal and Σ is diagonal any image can be decomposed by:

$$V^T = \Sigma^{-1}U^TX$$

where U the set of eigenfaces (each is one column), and Σ are the singular values. Some of the eigenfaces are shown below:

Fig. 3 *Eigenfaces*
The faces are Eigenface #2, Eigenface #4 Eigenface #9.

So we store U and Σ as the model, and V transposed as the features of training set. We can decompose any image to V transposed, and use that as the image's feature.

MASK DETECTOR

To detect whether the face is wearing a mask, we use a extraction method dedicated in object detection to extract features — Histogram of Oriented Gradients (HOG). The features are fed into an SVM to judge whether the face is wearing a mask.

The result shows that it is a simple task to determine whether the face is wearing a mask: linear SVM still produces 100% accuracy.

CLASSIFIER

After feature extraction using PCA, we use V transposed in PCA and labels of the belonging of the faces to train an SVM. We calculate its confidence level on 50-class-classification (information entropy or maximum probability) to determine whether the face belongs to a person in dataset. Expert level T2 is now accomplished.

PCA, SVM and maximum probability together yields ROC AUC of 0.91, while measuring entropy as confidence level yields a lower ROC AUC, which is about 0.90.

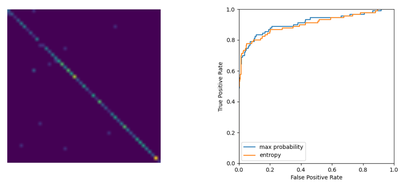


Fig. 4 *Confusion Matrix of the Classifier and the ROC curve*

WAY TO THE FINAL MODEL

After we directly implement the eigenface model for masked classification (Expert T3), we find out that its accuracy, 0.74, is not satisfying. Therefore, we tried some other approaches than eigenface decomposition.

Sparse Face Representation (SFR)

We tried to recreate the model on the paper *Robust Face Recognition via Sparse Representation*[1]. The paper declares it can obtain 90% accuracy on its dataset even if 40% of face is occluded.

SFR assumes that each face in one class can be represented by the linear combination of the other faces in this class. Consequently, the faces which the classifier has seen before has a sparse representation in feature spaces, whereas the other faces do not satisfy the property.

The task of detecting is now convert into solving the coefficients of the representation. We use second-order cone programming to compute the coefficients, and calculate the sparse concentration index to judge whether the face can be represented sparsely.

Despite its great performance on the dataset used on the paper, we do not achieve similar performance in our dataset. The accuracy is only 60%, which is even lower than vanilla eigenface methodology.

We discover that our dataset consists of faces with diverse facial expressions and poses, while the dataset on the paper only has variations on light condition. The former introduces more non-linear property of one class, and the linear assumption cannot be satisfied well.

Eyes Feature Extraction

Inspired by Apple's face recognition with mask, we tried to create a model focusing on the feature near eyes instead of whole obscured face.

After we get the feature points on the face, we extract the bounding box of eyes as the input. Then, we do similar PCA to extract "eigen-eyes" and each image's feature. Finally, we train an SVM based on these features.

Unfortunately, this model, which yields an accuracy of only 58%, does not work appropriately as well. We think that the images in dataset do not provide enough resolution for the images to be distinctive. What is worse, many faces are wearing glasses, whose reflection make the features vaguer. The eigen-eyes below shows that there are few distinctive features extracted.

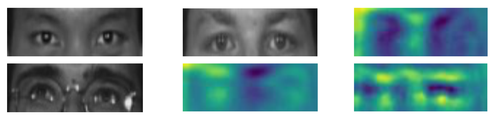


Fig. 5 *Extracted Eyes and Eigen-eyes*

[1]: J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210-227, Feb. 2009, doi: 10.1109/TPAMI.2008.79.

UNIFIED MODEL

After tried some other approaches than eigenface decomposition, we find out that they do not perform even as well as the original eigenface model. However, the eigenfaces of the masked dataset is wasting a lot of resourses on identifying the differences of masks, and the facial features are also mixed up with masks.

Fig. 6 *The eigenfaces of masked dataset is trying to analyze the mask a lot*

Therefore, we decided to dive deeply into the PCA process, and continue using the eigenfaces of unmasked dataset, even if we are detecting the faces with masks.

A problem immediately occurs: if we directly decompose the masked face using $V^T = \Sigma^{-1}U^TX$, the features will try to imitate the mask with whiter faces, and thus yields bad results. The recomposed image is showed below:

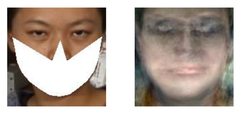


Fig. 7 *Masked image is tried to be recomposed by unmasked eigenfaces*

To fix this problem, we introduce importance weights M into the eigenfaces, and try to tell the eigenfaces not to imitate the part which the weight is low. An intuitive approach is to do element-wise product $U' = M \otimes U$, because that resembles only decomposing the high-weighted part and ignoring the low-weighted part.

However, we thought a little deeper and find that there is no difference between this approach and imitating a black mask. The reason is that the weighted eigenfaces U' is no longer orthogonal, so the decomposition of some component is done more than once.

There are two way to solve this: making U' orthogonal by Gram-Schmidt Orthogonalization, and try to decompose a face on non-orthogonal basis. The former is not only much more time-consuming, but also not lostless. So we choose the latter.

Since the singular values of the eigenfaces are decreasing by order, a greedy algorithm naturally comes out: If the face has some component which can be decomposed on both former and latter weighted eigenface, it will decompose all of it on the former.

Formally, the process is written in pseudo code as below:

1. Set $i := 0$.
2. For a masked face x from X, compute its projection on i-th weighted eigenface: $p_i = \frac{\langle x, M \otimes u_i \rangle}{\|M \otimes u_i\|}$
3. Subtract the component from x: $x := x - p_i(M \otimes u_i)$
4. The projection should be divided by its correspondent singular value to fit in v: $v_i := p_i/\sigma_i$
5. Set $i := i + 1$, go to 2 if i is less then the number of eigenfaces.

It turned out that this methodology functions as we want. We can recompose the entire face by $X = U\Sigma V^T$, where U is unweighted eigenfaces.

Fig. 8 *From left to right: Original Masked Image; Recomposed image using naive SVD Decomposition; Repomposed image using EIW Decomposition*

We call this method *Eigenface with Importance Weights Decomposition (EIW Decompositoin)*. As the figures show, the estimated V of a masked face resembles to that V in corespondent unmasked face. Therefore, we can train the SVM with only unmasked data as well, because the masked and unmasked version of a face have similar features in the context of EIW Decomposition.

Thus, we fulfill not only T3 in expert level, but also the bonus level by changing the SVM training process.

The model can be further unified to work on all tasks in one training procedure for Expert T4:

1. Do PCA and fit SVM on unmasked faces;
2. For an income image, use the mask detector build before to detect whether a mask exists:
   - If so, do EIW decomposition and feed to the SVM
   - If not, do SVD decomposition and feed to the SVM.

PERFORMANCE

Our new unified model yields exceptional performance on T3. With the Eigenfaces trained on unmasked data and SVM trained on masked data, it produces ROC AUC of 0.89, that is really close to the models' recognizing unmasked faces. On bonus level, it also produces 76% of accuracy in 50-class classification. The time performance, nevertheless, decreases due to non-parallel procedure.

For the combined pipeline in expert level T4, we also get an accuracy of 83.6%.

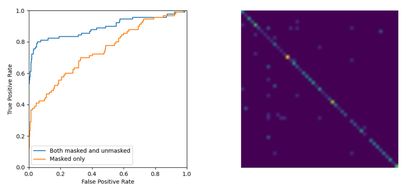


Fig. 9 *ROC curve of different method in Expert T3; confusion matrix of Bonus level*

The left figure shows the significant improvement of EIW decomposition than naive SVD in expert T3. The ROC AUC of EIW is 0.89, but only 0.74 for naive SVD.