Project 2: Αναγνώριση προσώπου από Egenfaces

Σε αυτό το project, μαθαίνουμε την εφαρμογή του *principle component analysis* για κατηγοριοποίηση . Τα αντικείμενα που θα κατηγοριοποιήσουμε είναι τα πρόσωπά μας.

Περιεχόμενα

0. Δεδομένα προσώπου

Οι εικόνες είναι 2D arrays. Το κλασικό PCA ωστόσο δουλεύει σε διανυσματικά δεδομένα, όπως έχουμε δει στο παράδειγμα Ύψος-Βάρος στα εργαστήρια PCA και SVD. Οπότε πρίν την ανάλυση, οι 2διάστατες εικόνες πρέπει να μετασχηματιστούν σε long 1D διανύσματα. Αυτό γίνεται απλά συνθέτοντας τις n στήλες του πίνακα *m by n*. Στο MATLAB, αυτή η μετατροπή σε διάνυσμα για έναν πίνακα **A** είναι απλά `a = A(:)`.

Τρεις είναι οι διαθέσιμες βάσεις δεδομένων: att, Yale και Konstanz. Οι δύο πρώτες είναι ελεύθερα διαθέσιμες στο Διαδίκτυο, το τελευταίο ήταν που δημιουργήθηκε από τις εικόνες των συμμετεχόντων του μαθήματος. Μπορείτε να καθορίσετε τη βάση δεδομένων, με την οποία θα θέλατε να εργαστείτε, στη μεταβλητή "db":

```
db = 'att.mat';
db = 'yale.mat';
db = 'konstanz.mat';
```

Για να περάσετε τα δεδομένα προσώπου στο MATLAB τρέξτε την `load(db)`. Στο χώρο εργασίας, θα εμφανιστούν τρεις μεταβλητές:

- `F` - ο πίνακας Face, στις γραμμές του οποίου βρίσκονται οι διανυσματικές εικόνες των προσώπων;
- `N` - ο αριθμός των ατόμων (προσώπων) στη βάση δεδομένων;
- `m, n` - το μέγεθος του πίνακα εικόνων προσώπου (πρίν την διανυσματικοποίηση).

```
db = 'konstanz';
load(db);
```

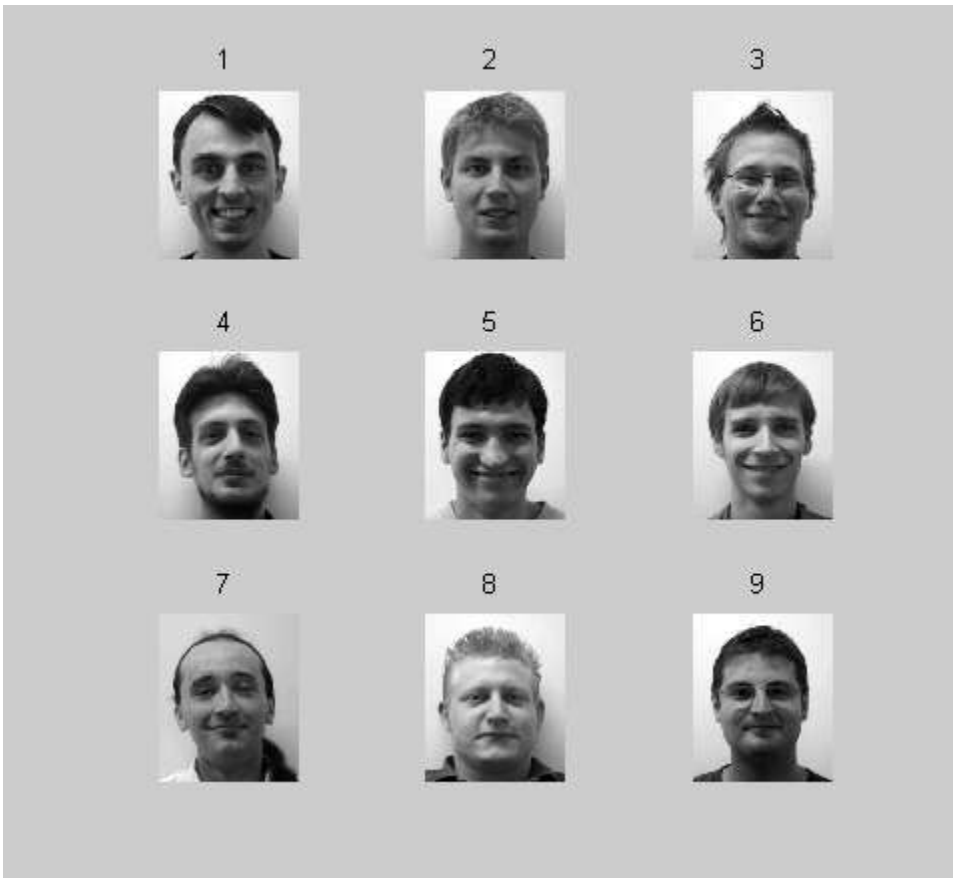
Εξερευνώντας τα πρόσωπα:

Για κάθε αντικείμενο (πρόσωπο), συλλέχθηκαν διάφορες εικόνες που αντιστοιχούν σε διαφορετικές εκφράσεις προσώπου. Μπορείτε να τις διερευνήσετε με την ένδειξη της exprNum μεταβλητής σε αυτό το cell:

```
NN  = size(F,1);        % Overal number of face images.
fps = NN/N;             % Number of faces per subject.

% Display faces
exprNum = 2;            % The number of the particular face expression.

nrows = round(sqrt(N));
ncols = ceil(N/nrows);
figure(1); clf; set(gcf, 'Name', 'Faces');
for ii=1:N
    subplot(nrows,ncols,ii);
    tmp = (ii-1)*fps + exprNum;
    imagesc( reshape(F(tmp,:),m,n) );
    colormap gray; axis equal tight off;
    title(num2str(ii));
end
```



1. PCA

Όπως συχνά συμβαίνει στα εφαρμοσμένα μαθηματικά, μας ενδιαφέρει μια καλύτερη εκπροσώπηση των δεδομένων μας, δηλαδή σε μια καλύτερη βάση (από την στάνταρ). Υποθέτουμε ότι υπάρχει κάποιου είδους γραμμική εξάρτηση μεταξύ των διανυσμάτων προσώπων και έτσι τα *principle axes* που παίρνουμε από το PCA θα πρέπει να παρέχουν μια αρκετά καλή βάση. Στη συνέχεια, θα ελέγξουμε αυτή την υπόθεση.

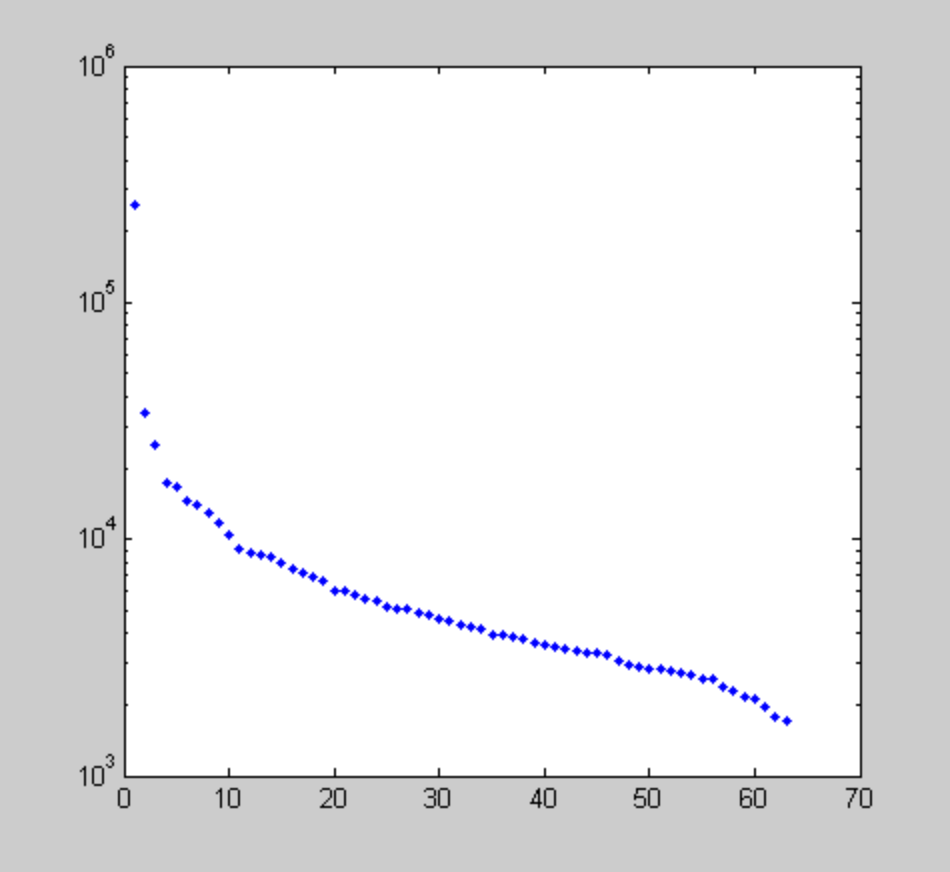
Εργασία 1: SVD

- Υπολογίστε το SVD του πίνακα προσώπων, **F**;
- Χαράξτε τα singular values σε λογαριθμική κλίμακα. Μειώνονται αρκετά γρήγορα;
- Οι στήλες του πίνακα SVD **V** είναι τα *principle axes* για τα δεδομένα προσώπων. Υπολογίστε τα κύρια στοιχεία (συντεταγμένες) των εικόνων προσώπων w.r.t και αποθηκεύσετε σε σειρές ενός πίνακα **PC**.

Προσοχή: οι εντολές MATLAB `svd()` και `svd( ,0)` θα υπολογίσουν τετραγωνικούς πίνακες **V** μεγέθους αρκετών χιλιάδων. Είναι περιττό, δεδομένου ότι πρόκειται να χρησιμοποιήσετε μόνο πολλές πρώτες στήλες του **V**. Χρησιμοποιήστε την εντολή `svd(,'econ')` για τον υπολογισμό μόνο των πρώτων *N* στηλών του **V**.

YOUR CODE HERE:

```
[U S V] = svd(F, 'econ');
PC = U*S;                                   % the principle components
figure(2); clf; semilogy(diag(S), '.');
```



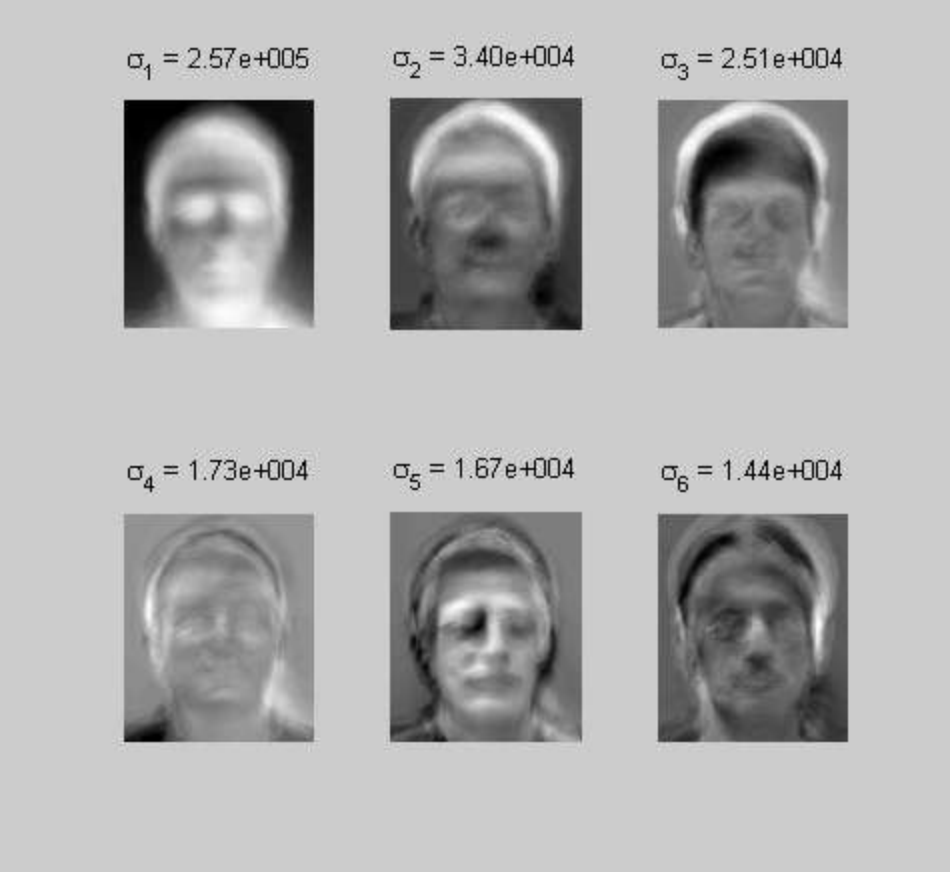
Εργασία 2: Εξερευνώντας το "eigenfaces"

Η νέα βάση, τα *principle axes*, έχουν την ίδια διάσταση με τις αρχικές εικόνες προσώπων και, συνεπώς, μπορεί να αναδιαμορφωθεί και να εικονοποιηθεί ώς πίνακας. Τα πρώτα 6 eigenface εμφανίζονται παρακάτω. Μπορείτε να αναγνωρίσετε τον εαυτό σας σε αυτές;

- Φτιάξτε ένα plot με eigenfaces μόνοι σας.
- Αναρωτηθείτε, γιατί τα eigenfaces έχουν αυτή την προσωποειδή εικόνα;

Ο ΚΩΔΙΚΑΣ ΣΑΣ ΕΔΩ:

```
nrows = 2;
ncols = 3;
figure(3); clf; set(gcf, 'Name', 'EigenFaces');
for ii=1:nrows*ncols
    subplot(nrows,ncols,ii);
    imagesc( reshape(V(:,ii),m,n) );
    colormap gray; axis equal tight off
    title(['\sigma_' num2str(ii) ' = ' num2str(S(ii,ii), '%1.2e') ]);
end
```



2. Κατασκευή του Eigenfaces subspace

Τα eigenfaces διαμένουν στο N-διαστάσεων τμήμα όπου βρίσκονται τα δεδομένα προσώπων. Παρόλα αυτά, σύμφωνα με την υπόθεση της γραμμικής εξάρτησης, ο πολύ λιγότερος αριθμός των διανυσμάτων βάσης πρέπει να δώσει μια λογική αναπαράσταση των δεδομένων προσώπων. Ας προσπαθήσουμε να μειώσουμε τον αριθμό των eigenfaces από N σε nDim και να δούμε αν τα διανύσματα βάσης nDim δίνουν όντως μια ικανοποιητική επανακατασκευή.

```
nDim = 20;                  % dimensionality of the eigenfaces subspace
V1 = V(:,1:nDim);           % reduced basis of eigenfaces
```

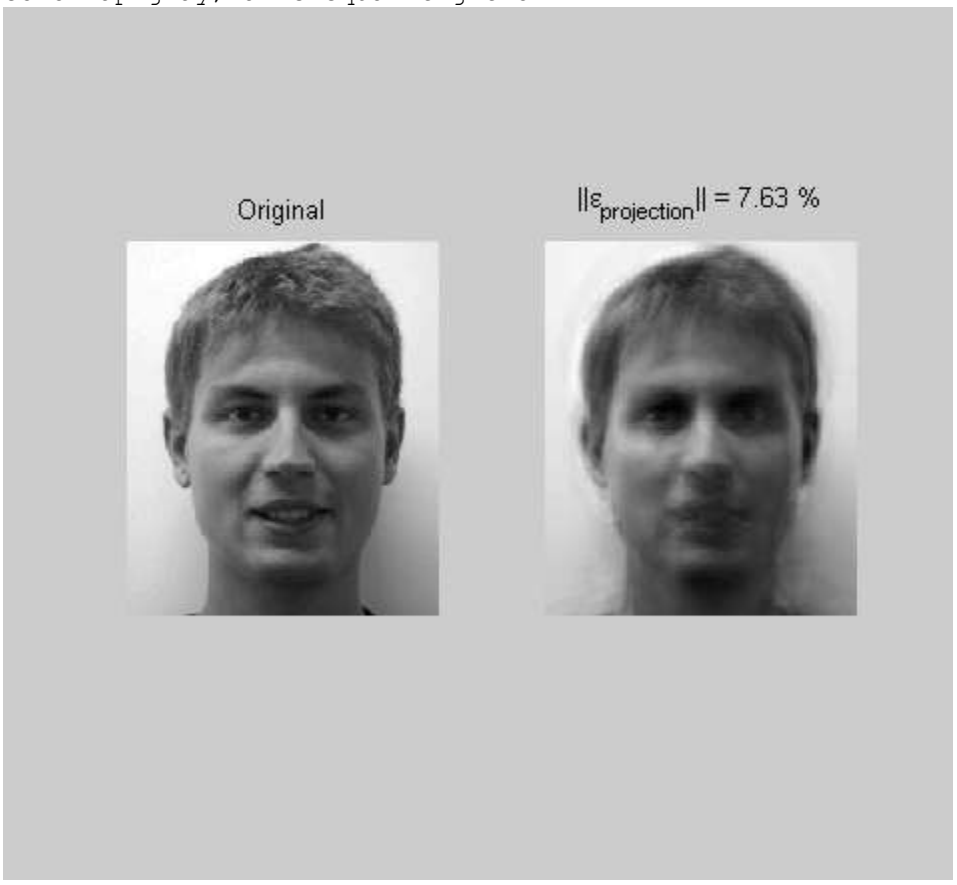
Εργασία 3: Εξερευνήστε τις ανακατασκευές

Διαλέγουμε ένα τυχαίο πρόσωπο `f` από τον πίνακα προσώπων **F**.

- Από τον πίνακα **PC**, εξάγετε τις πρώτες αρχικές συντεταγμένες `nDim` που αντιστοιχούν στο `f` και υπολογίστε την ανασυγκρότηση, `f1`;
- Υπολογίστε το σχετικό λάθος rms της ανασυγκρότησης, rmse;
- Όταν γίνει, τρέξτε το cell και παρατηρήστε την ποιότητα της ανασυγκρότησης.

```
fn = randint(1,1,[1, NN]);            % Select face at random
f   = F(fn,:)';
% YOUR CODE HERE:
pc1 = PC(fn,1:nDim);                  % project it on the first nDim principle
axes
f1 = V1*pc1';                         % reconstructed face
rmse = norm(f-f1)/norm(f);            % rms reprojection error

% Plot the original and reconstructed images
figure(4); clf;
subplot(1,2,1); imagesc(reshape(f,m,n)); title('Original');
axis equal tight off
subplot(1,2,2); imagesc(reshape(f1, m,n));
title(['||\epsilon_{projection}|| = ' num2str(rmse*100,'%2.2f') ' %']);
colormap gray; axis equal tight off
```



3. Κατασκευή του feature space

Μέχρι στιγμής, κάθε θέμα συνδέεται με πολλά διανύσματα nDim των κύριων συστατικών. Στόχος μας είναι να καθορίσουμε ένα ενιαίο χαρακτηριστικό "feature vector" για κάθε πρόσωπο. Με τον πιο απλό τρόπο, το εν λόγω διάνυσμα μπορεί κατασκευαστεί απλά παίρνοντας τον μέσο όρο των αντίστοιχων principle component διανυσμάτων. Αυτά τα `N nDim`-feature διανύσματα καθορίζουν το feature space που θα γίνει αναζήτηση από τους αλγόριθμους αναγνώρισης προσώπων. Παρατηρήστε την τεράστια μείωση διαστάσεων που σημειώθηκε κατά τη μετακίνηση από το μ-φορές-n-διαστάσεων χώρο όλων των εικόνων στον nDim διαστάσεων χώρου χαρακτηριστικών (feature space).

Δεν χρειάζεται να υπολογίσετε το διάνυσμα (feature vector) μόνοι σας. Ο κώδικας κάτω το κάνει για εσάς. Τα διανύσματα που προκύπτουν αποθηκεύονται στις σειρές του `N-by-nDim` πίνακα PC_mean.

Το σχήμα δίπλα δείχνει τις ανασχηματισμένες εικόνες προσώπου από τα διανύσματα χαρακτηριστικών (feature vectors).

```
PC_mean = zeros(N, nDim);       % average feature vectors for each subject
for ii=1:N
    ix1 = fps*(ii-1)+1; ix2 = ix1+fps-1;
    PC_mean(ii,:) = mean(PC(ix1:ix2,1:nDim));
end
% Plot faces reconstructed from the feature vectors
nrows = round(sqrt(N));
ncols = ceil(N/nrows);
figure(1); set(gcf, 'Name', 'Face averages');
for ii=1:N
    subplot(nrows,ncols,ii);
    imagesc(reshape(V1*PC_mean(ii,:)',m,n));
    colormap gray; axis equal tight off
    title(num2str(ii));
end
```

4. Αναγνώριση προσώπου

Αφού κατασκευάστηκε το feature space, ο αλγόριθμος για την αναγνώριση είναι απλός.

Για μία (διανυσματική) εικόνα προσώπου *f*:

- Υπολογίσατε το feature vector, δηλαδή τις συντεταγμένες στο (μειωμένο) eigenface basis;
- Υπολογίστε τις αποστάσεις μεταξύ των feature vectors που αντιστοιχούν στο *f* και τα αντικοίμετα στην βάση δεδομένων μας.
- Βρείτε το πλησιέστερο αντικείμενο.

Εργασία 4: Αναγνώριση προσώπου

- Να εφαρμοσθεί ο αλγόριθμος αναγνώρισης προσώπου που περιγράφεται επάνω για ένα πρόσωπο που έχει επιλεχτεί τυχαία για τεστ από τον πίνακα προσώπων μας.
- Σε μία εικόνα, απεικονίστε το τεστ και τα πρόσωπα που αναγνωρίστηκαν.
- Απεικονίστε τις αποστάσεις inter-feature-vectors σε ένα bar plot.

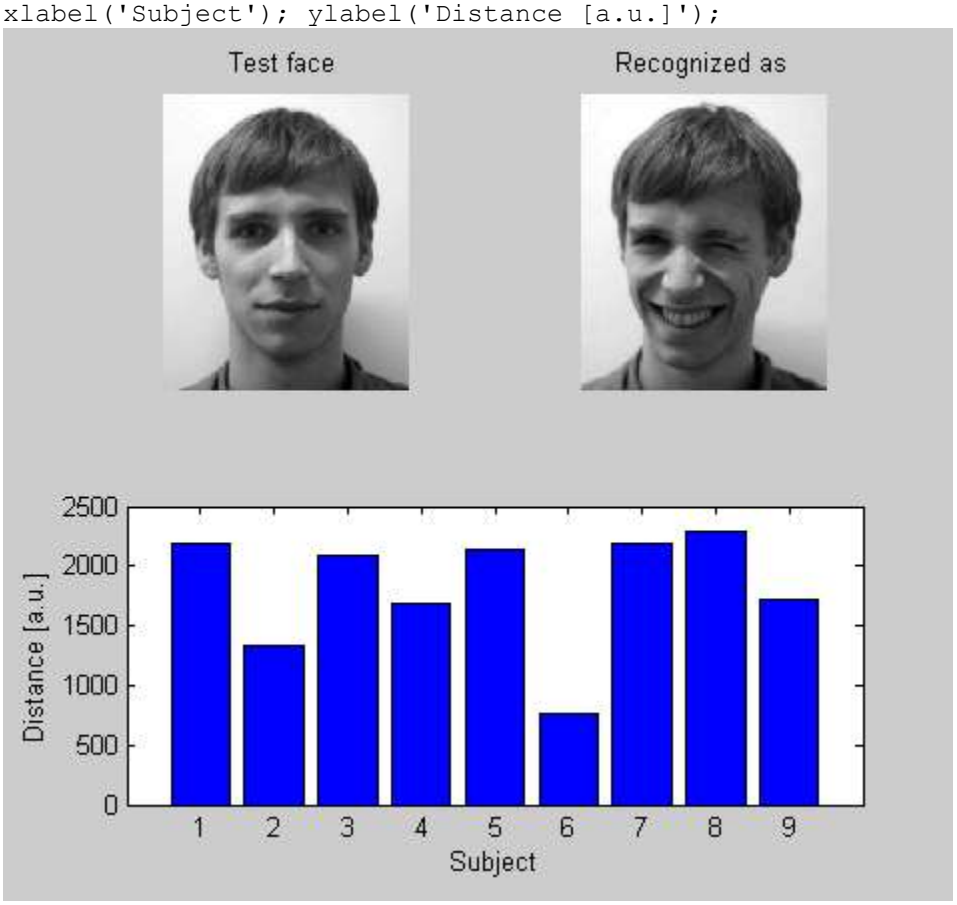
Το αποτέλεσμα θα πρέπει να μοιάζει με το παρακάτω:

Ο ΚΩΔΙΚΑΣ ΣΑΣ ΕΔΩ:

```
% Select a face at random and compute its feature vector:
fn = randint(1,1,[1, NN]);
f = F(fn,:);                          % test face
pc1 = V1'*f';                         % its principle component

% Compute distances in the feature space
Dist = PC_mean - ones(N,1)*pc1';
dist = sqrt(mean(Dist.^2, 2));
[min_dist, min_ix] = min(dist);

% Plot
figure(5); clf; set(gcf, 'Name', 'Face recognition')
subplot(2,2,1); imagesc( reshape(f,m,n) );
axis equal tight off; colormap gray;
title('Test face');
subplot(2,2,2); imagesc( reshape( F((min_ix-1)*fps+7,:), m,n) );
axis equal tight off; colormap gray;
title('Recognized as');
subplot(2,2,[3 4]); bar(dist, 'FaceColor', 'b');
xlabel('Subject'); ylabel('Distance [a.u.]');
```



5. Αναγνώριση προσώπου σε πραγματικό χρόνο

Στην πραγματικότητα, η αναγνώριση προσώπου με αυτόν τον απλό τρόπο δεν είναι πολύ αξιόπιστη. Παρατηρήστε τις αποστάσεις στο γράφημα παρακάτω:

```
camF = double(t);
camf = camF(:);

pc1 = V1'*camf;                          % its principle component
% Compute distances in the feature space
Dist = PC_mean - ones(N,1)*pc1';
dist = sqrt(mean(Dist.^2, 2));
[min_dist, min_ix] = min(dist);

% Plot
figure(5); clf; set(gcf, 'Name', 'Face recognition')
subplot(2,2,1); imagesc( camF );
axis equal tight off; colormap gray;
title('Camera image');
subplot(2,2,2); imagesc( reshape( F((min_ix-1)*fps+5,:), m,n) );
axis equal tight off; colormap gray;
title('Recognized as');
subplot(2,2,[3 4]); bar(dist, 'facecolor', 'b');
xlabel('Subject'); ylabel('Distance [a.u.]');
```