EEL6935 Biometric Identification HW 4

Madhusudan Govindraju 39267182

Face Recognition using Eigenfaces

1.1 Find the Eigen Faces for corresponding top 10 eigen-vectors

1. The face images are loaded from the data sets and rearranged to form a vector of size 2500xM where M is the number of images. Each image was originally 50x50 which has been rearranges to a one dimensional vector 2500x1, This is Γ
2. Find the mean image Ψ
3. Subtract the mean image from every image to get the $\Phi(i)$ and we group all the $\Phi(i....m)$ together to get the matrix A.
4. Now if we find the covariance of A to find the eigenvectors and eigenvalues . The resulting matrix is very large, instead of finding the covariance matrix $A \times A^T$ we find the matrix $A^T \times A$ and then compute the eigenvectors(v_i)
$$A^T A v_i = \mu_i v_i$$
5. We sort the eigenvalues(μ) in descending order and then pick the top 10 eigenvectors corresponding to the top 10 eigenvalues, and calculate the eigenvectorsu_i of the AA^T matrix using the following formula,
$$u_i = A v_i$$
6. This u_i vector is rearranged in a 50×50 matrix to get the eigenfaces.
7. The weights can be calculated by multiplying the top N required eigenfaces with the matrix A.

1.2 Compute the eigen-coeff for top 30 eigen-faces

1. From the sorted vector containing the eigenvalues the top 30's corresponding *eigenvectors* are chosen.
2. Calculate the eigenvectors of AA^T with the formula $u_i = A v_i$.
3. The weights of the gallery images can be obtained from u_i with the formula $u_i^T \times A$. The weights for the probe images can be obtained using from u_i with the formula $u_i^T \times A_{probe}$.
4. We find the distance measure between these two weights to get the similarity matrix for the face recognition purpose.

1.3 ROC, CMC, Genuine and Imposter Disribution, Rank-1 Identification Curve

The above mentioned graphs are plotted for different coefficients taken from 30 to 100 in steps of 10. They can be found in the later sections of this report with appropriate headings. From the graphs it can be understood that increasing the number of coefficients do not have and effect on the performance of the system. This can be verified easily from figures 9 10 11 12.

1.4 Distance measures

Four different distance measures are used in this assignment they are
Eucledian Distance : $d_{st} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
Chebychev Distance : $d_{st} = max_j\{|x_{sj} - x_{tj}|\}$
Minkowski Distance : $d_{st} = \sqrt[p]{\Sigma_{j=1}^{n}|x_{sj} - x_{tj}|^p}$
City Block Distance : $d_{st} = \Sigma_{j=1}^{n}|x_{sj} - x_{tj}|$

Among these City - Block Distance measure seems to work the best having the rate at 85% and the worst distance measure would be chebyshev performing at 59% . The Distance measure in the order of performance is City-Block ,Euclidean, Minkowski, Chebyshev.

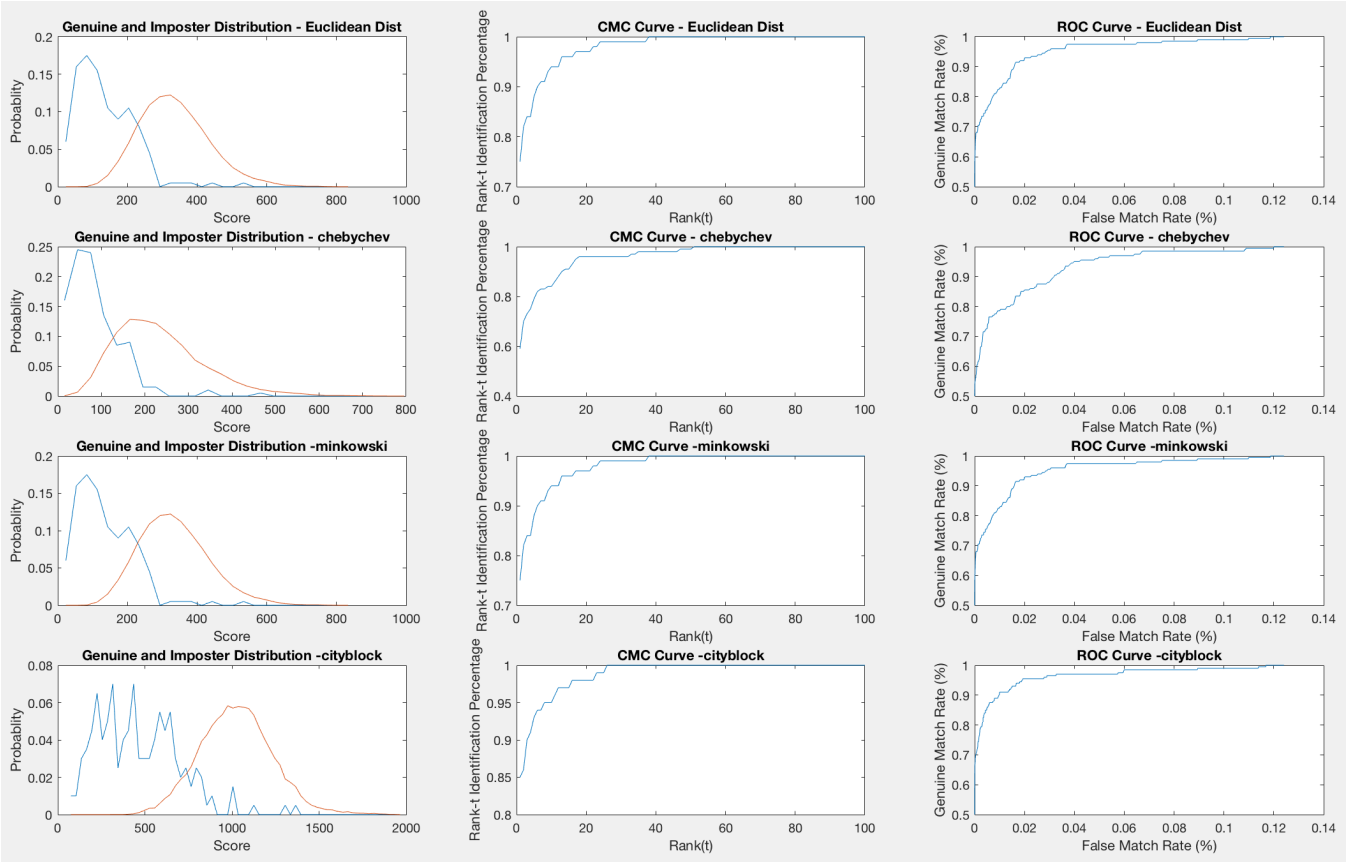

Figure 1: Plots for 30 Eigenvectors

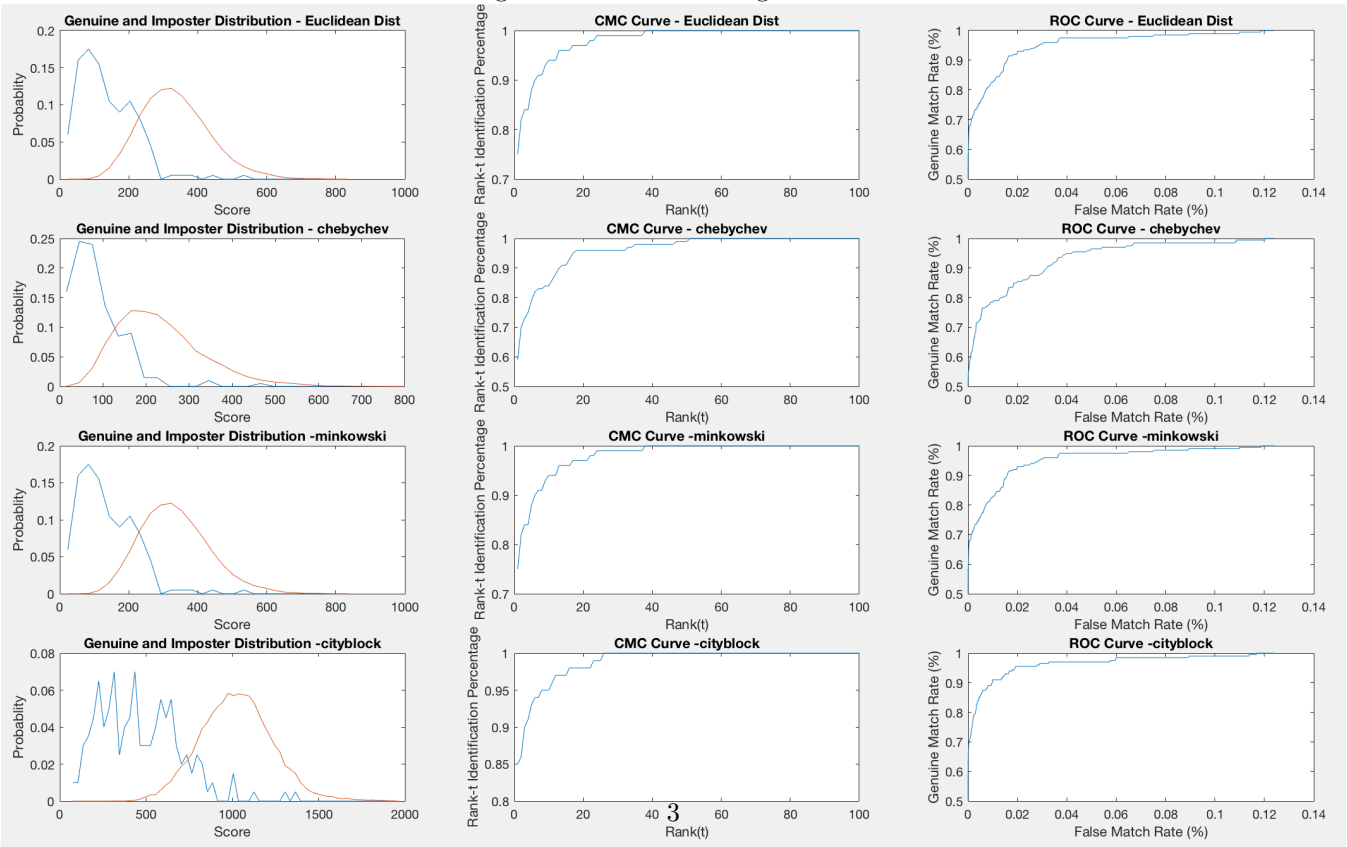

Figure 2: Plots for 40 Eigenvectors

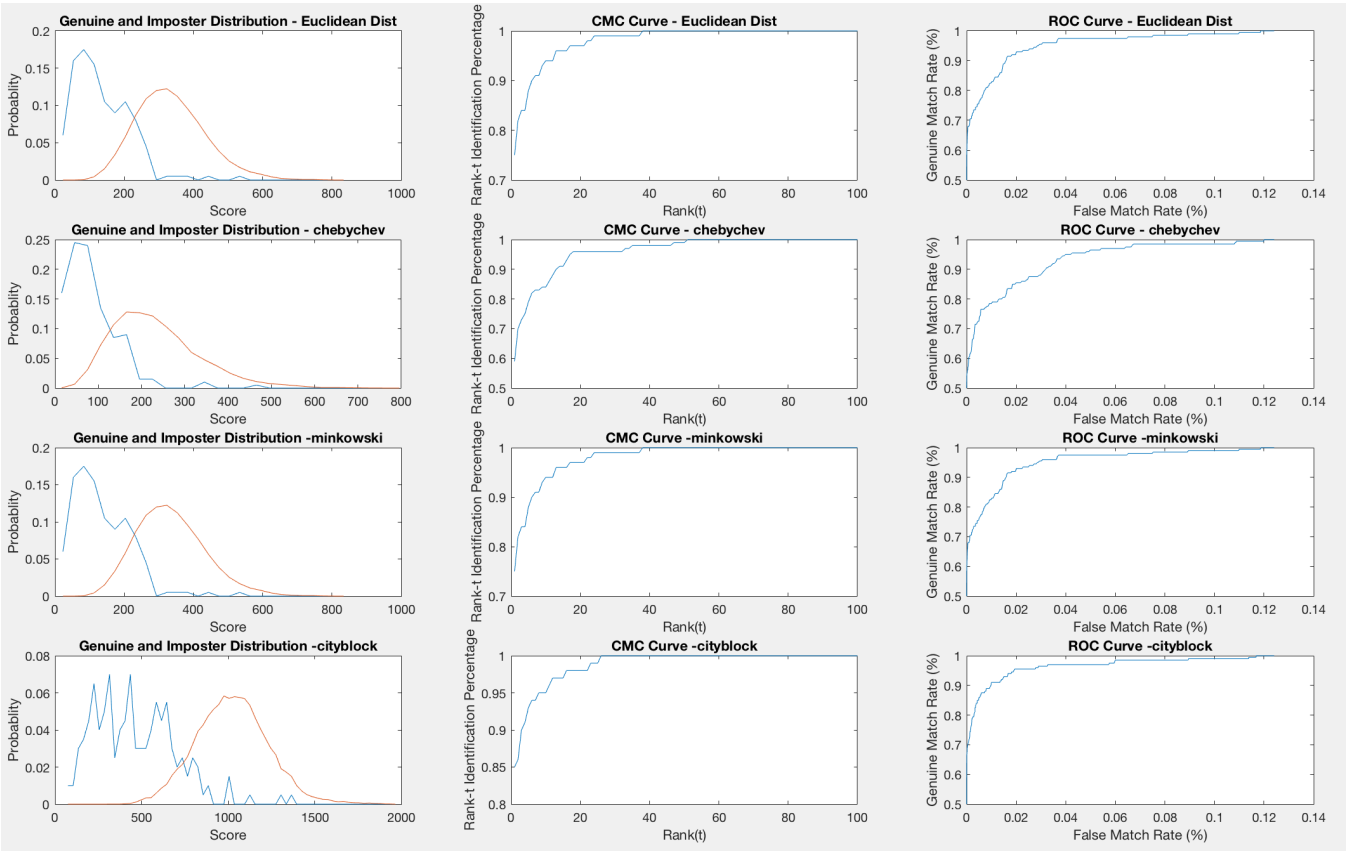

Figure 3: Plots for 50 Eigenvectors

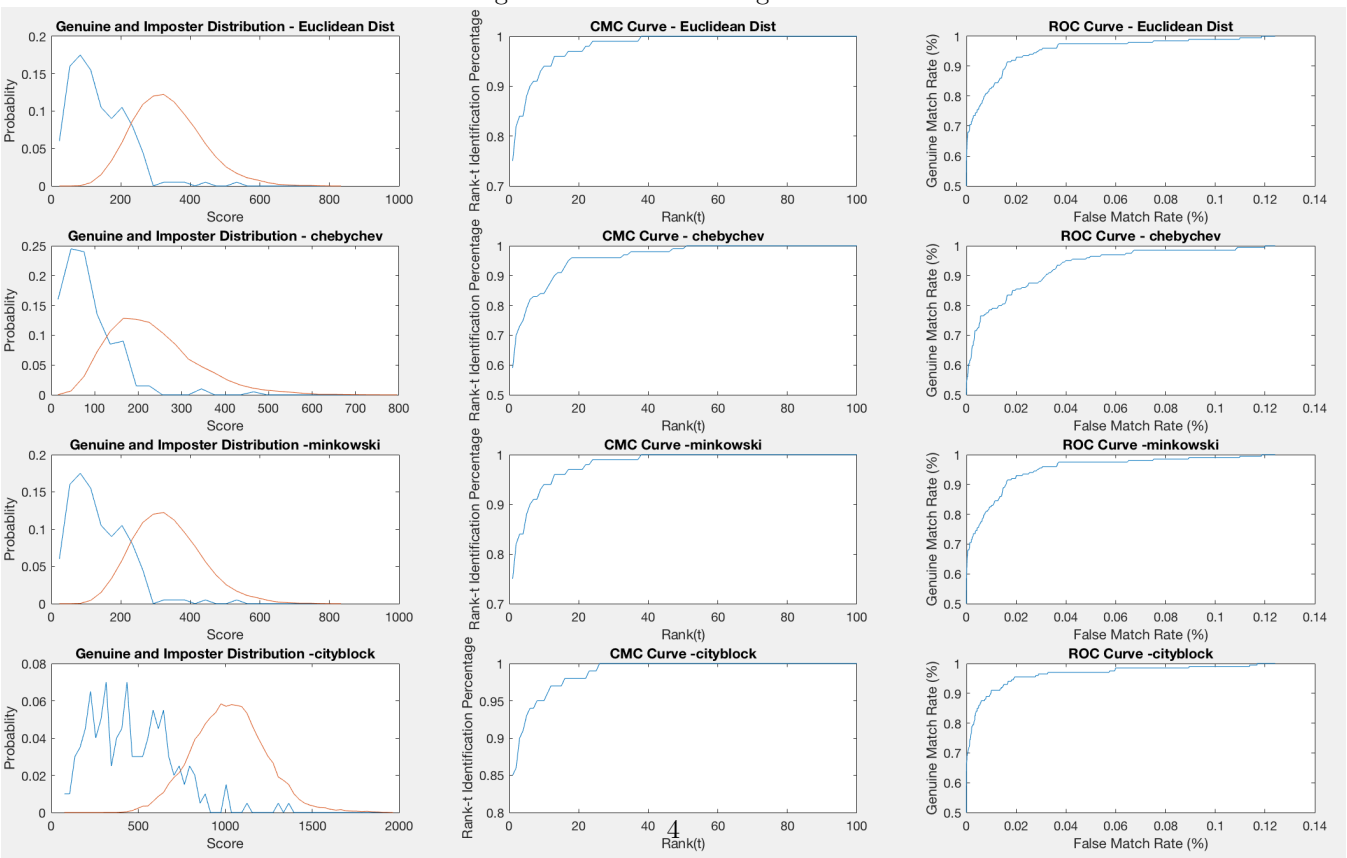

Figure 4: Plots for 60 Eigenvectors

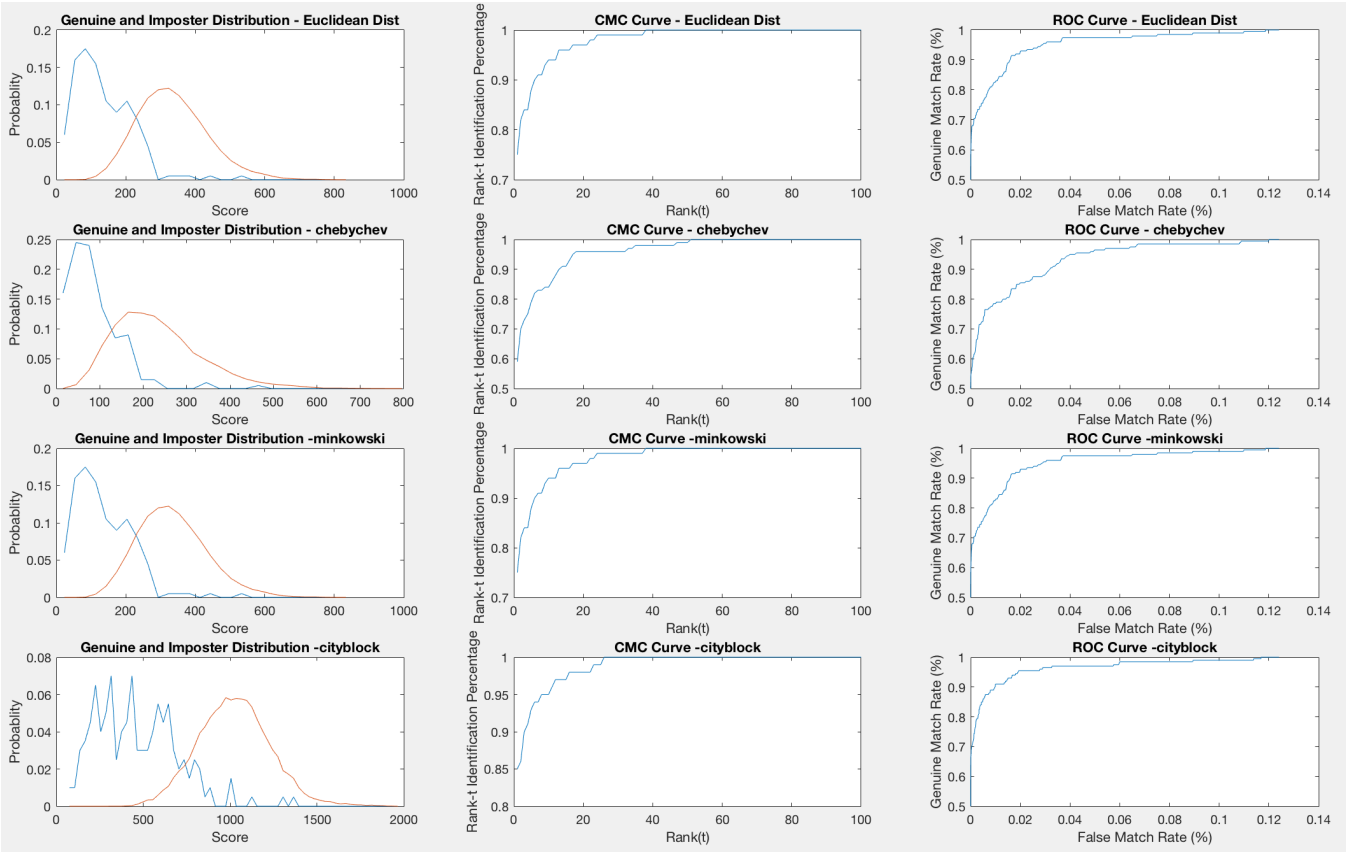

Figure 5: Plots for 70 Eigenvectors

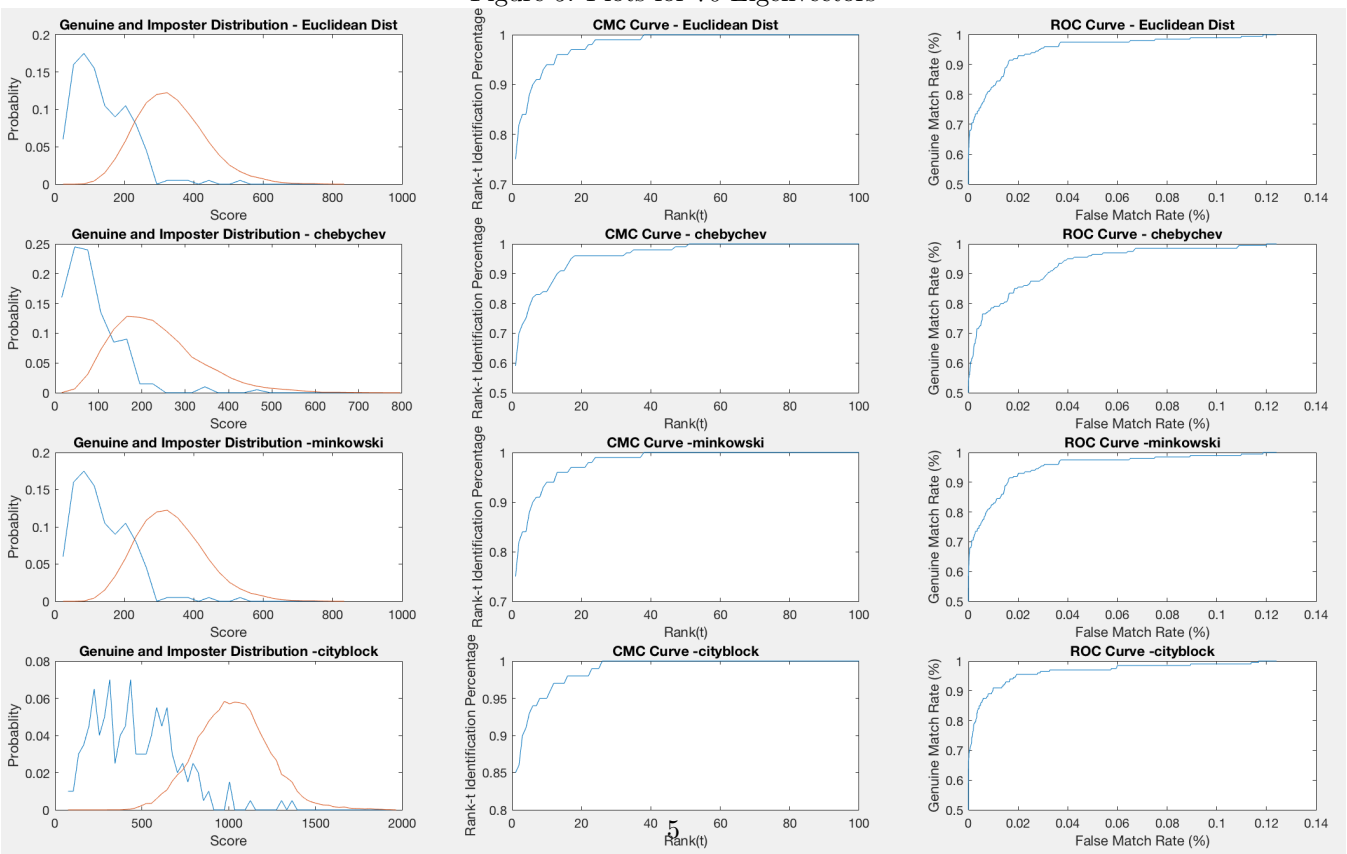

Figure 6: Plots for 80 Eigenvectors

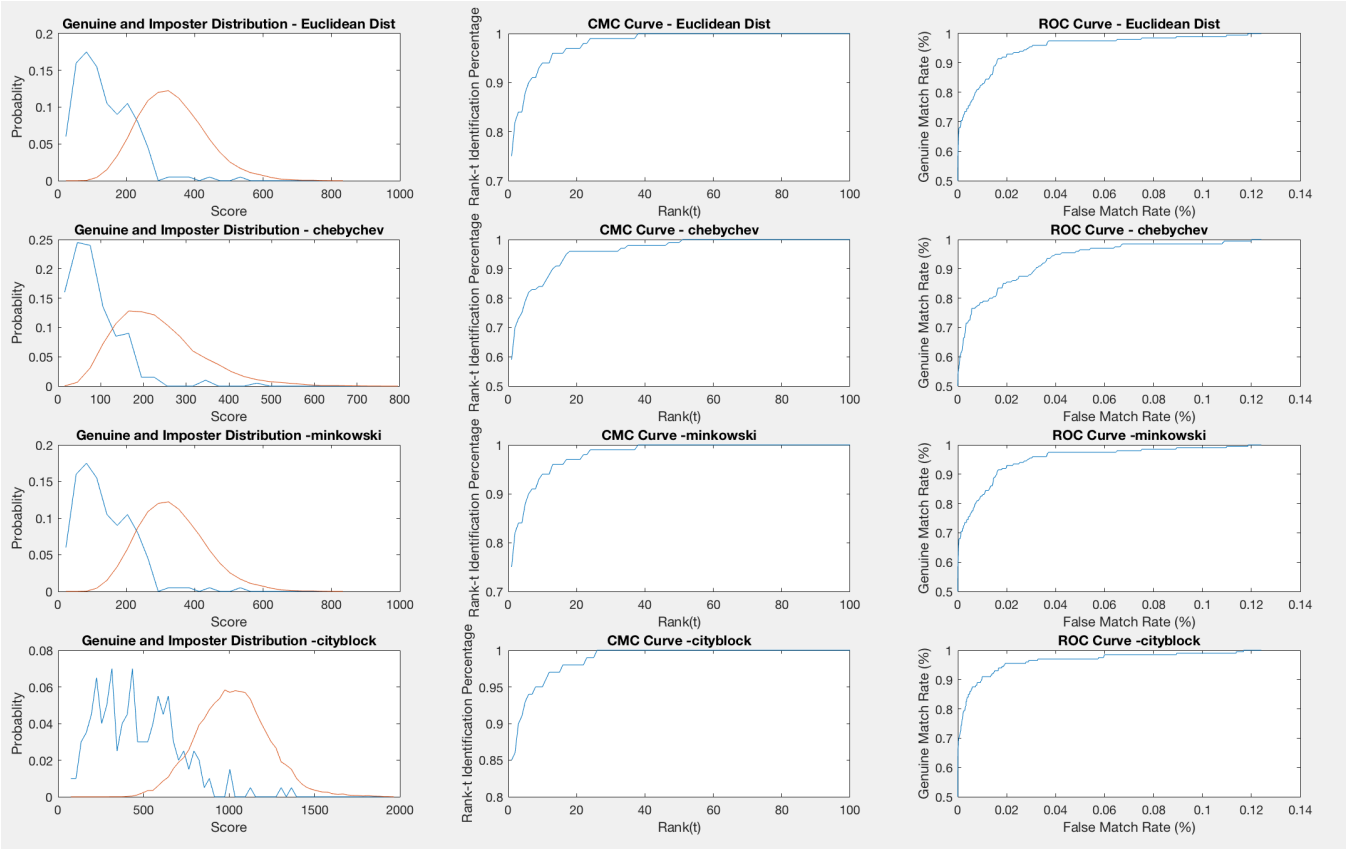

Figure 7: Plots for 90 Eigenvectors

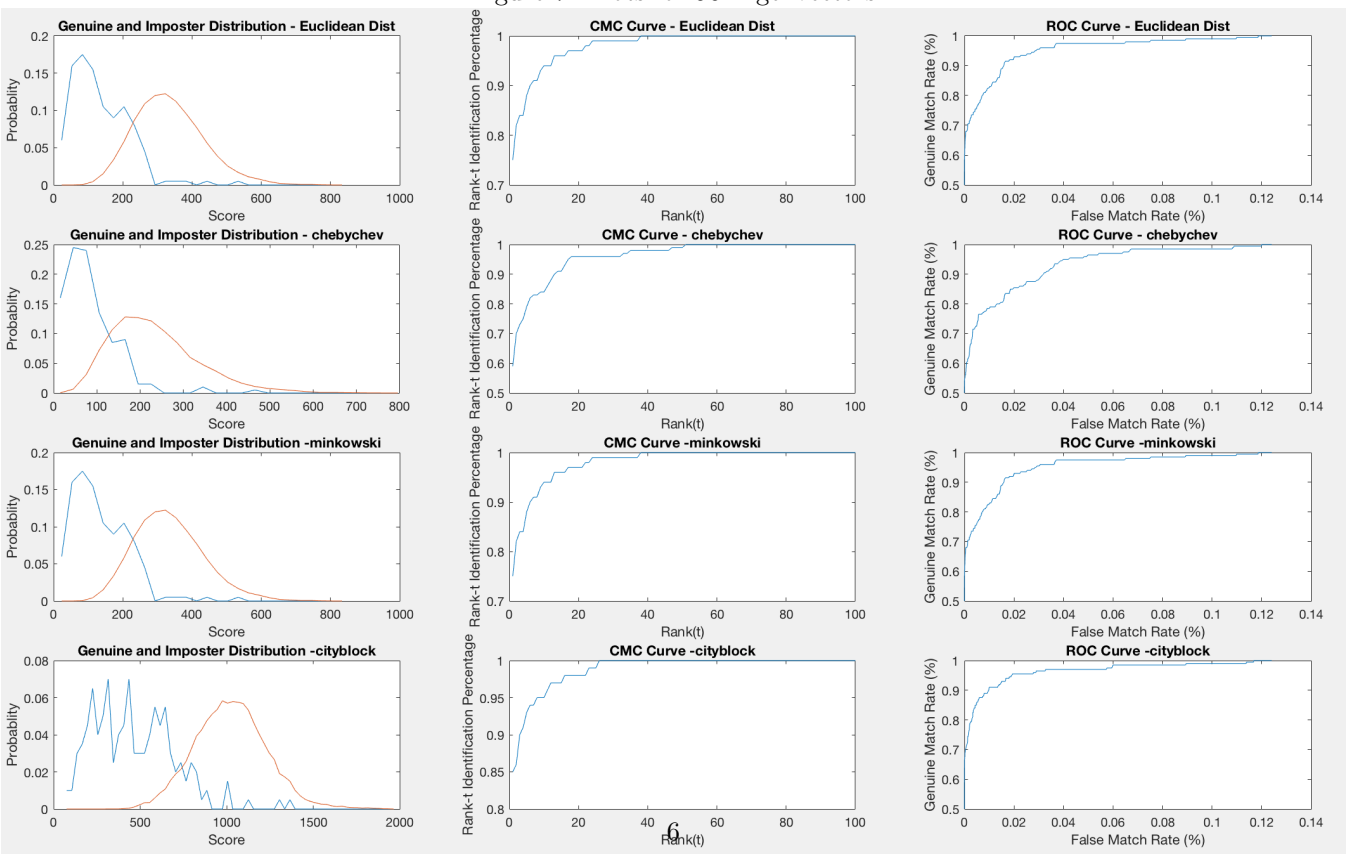

Figure 8: Plots for 100 Eigenvectors

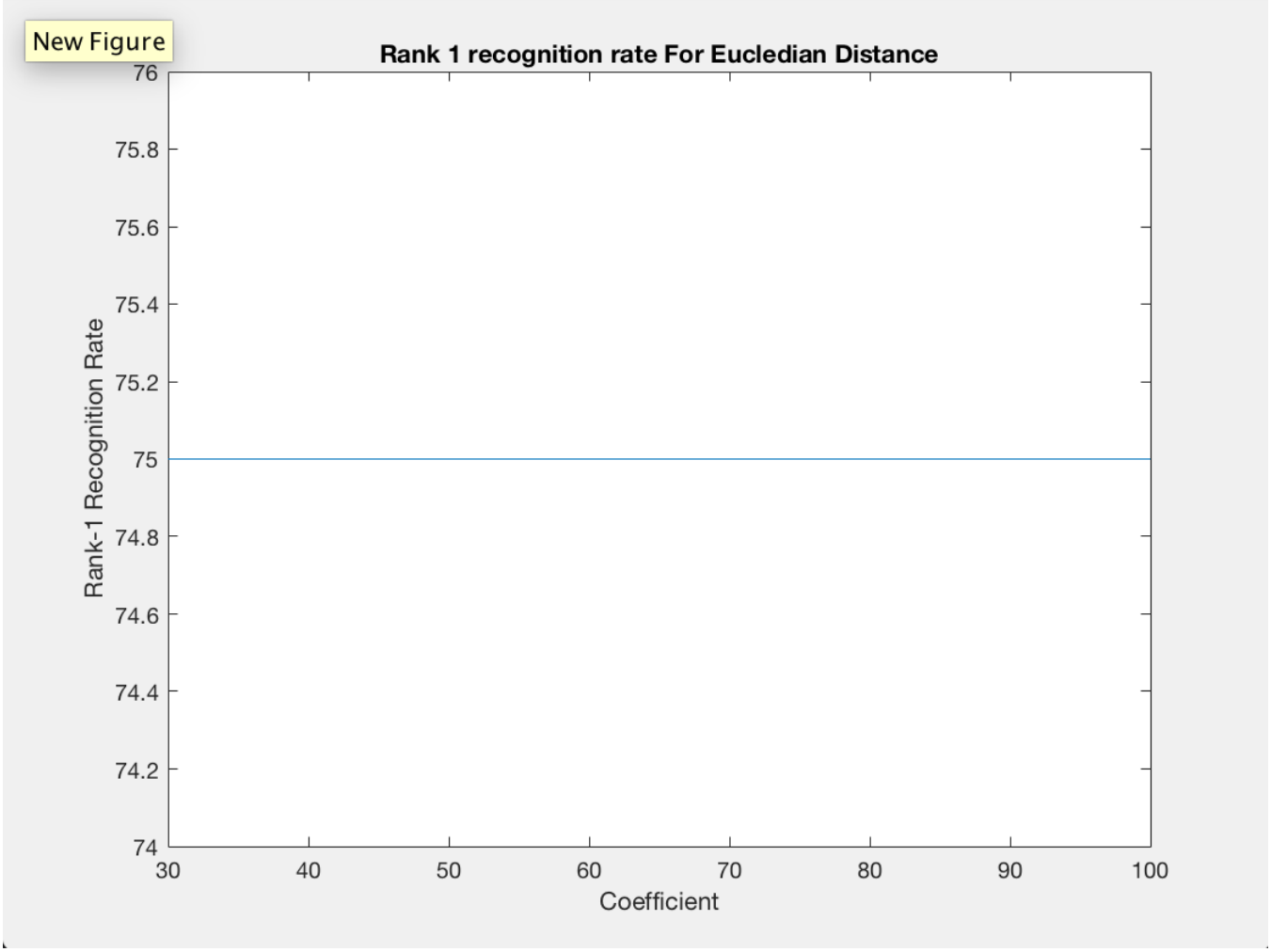

Figure 9: Rank 1 Recognition rate for Euclidean Distance

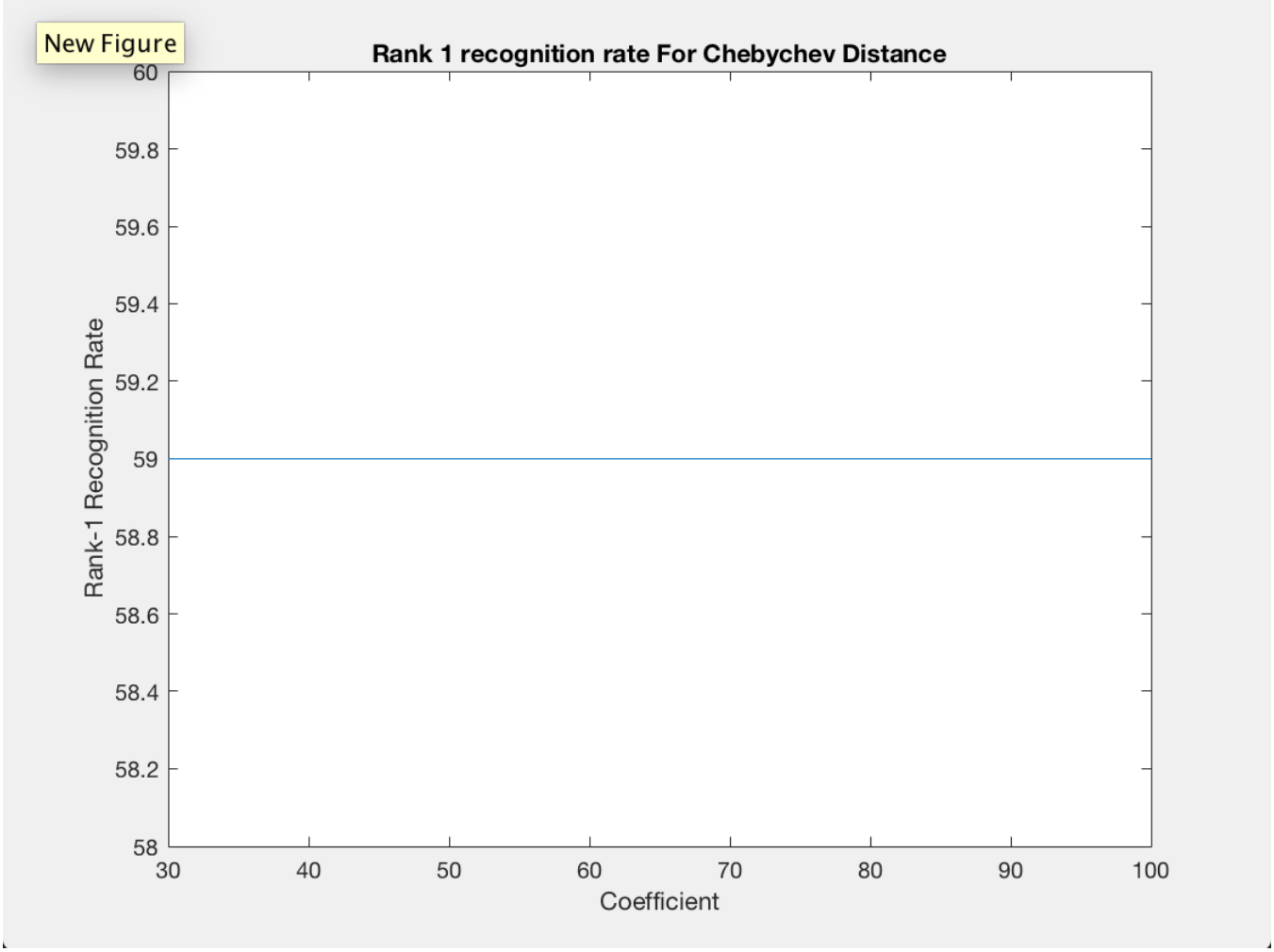

Figure 10: Rank 1 Recognition rate for Chebychev Distance

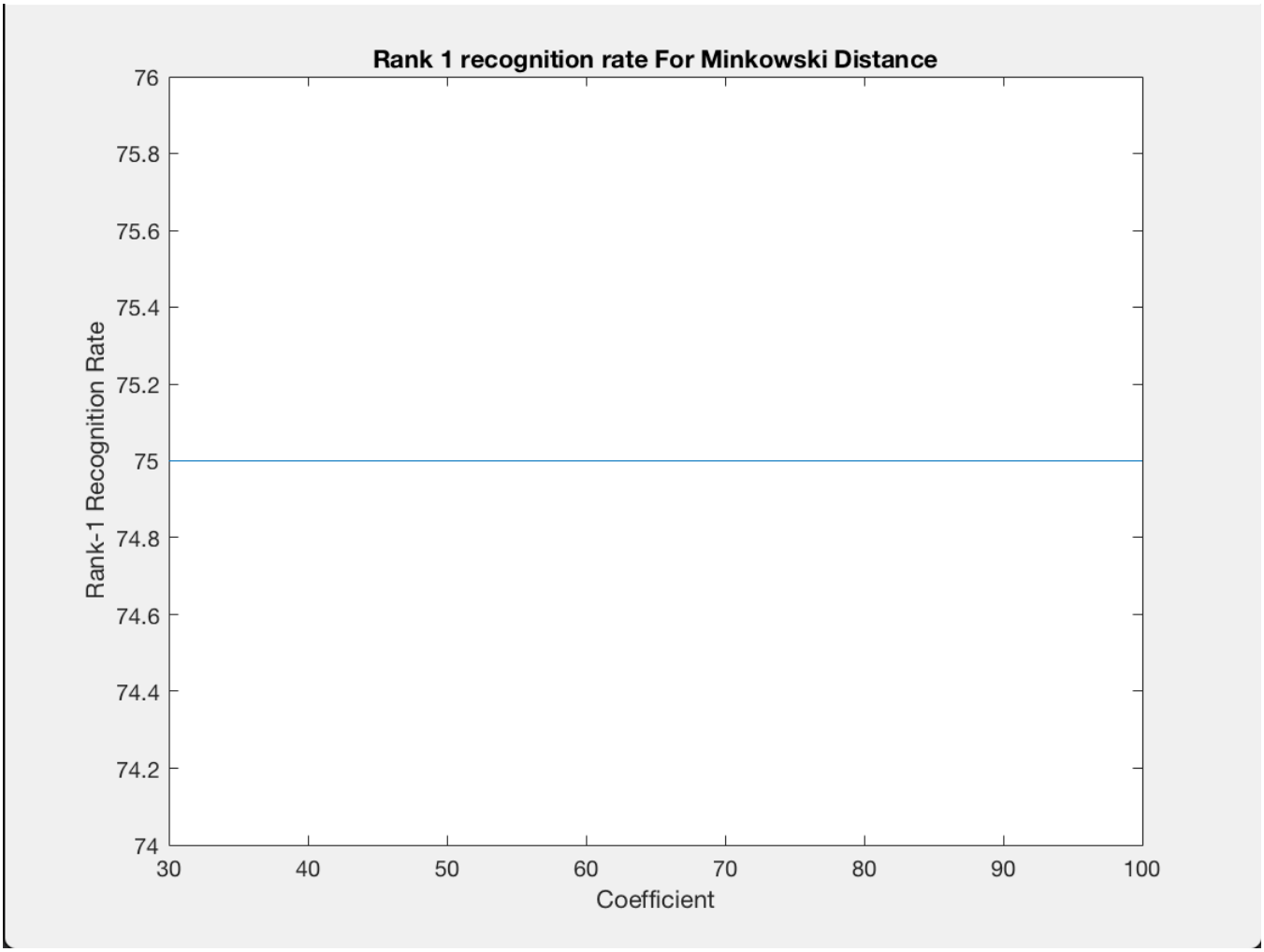

Figure 11: Rank 1 Recognition rate for Minkowski Distance

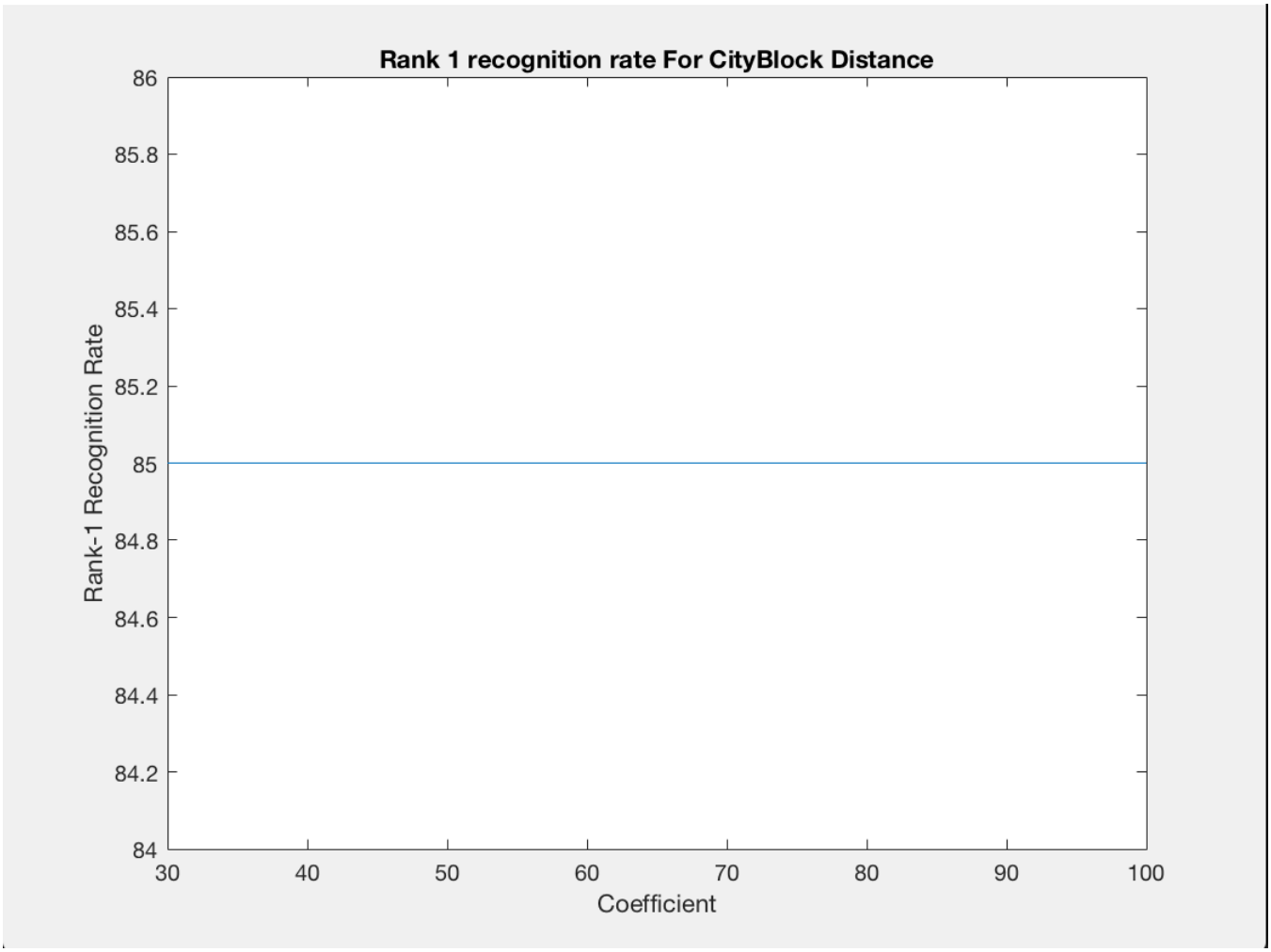

Figure 12: Rank 1 Recognition rate for City Block Distance