I. Introduction

In the last fifty years, there have been a lot technological advances in everything from mechanics to communication. Such development in technologies has rendered us the ability to perform tasks that would of been impossible fifty years ago. One of these is the possibility of tracking and recognizing 3D objects in an image or video. In which the computer is able to distinguish and identify a specified object in an image. This becomes helpful in industries where security is needed. For example, we can place a camera outside a building that is programmed to identify trespassers, people that shouldn’t be there. Therefore, for this project we are going to solve the problem of distinguishing between avatar and human faces. Where the given images are 75 x 50 pixels; therefore, the number of pixels is 3750 pixels per image.

II. Details of the Selected Algorithm

For this challenge, we are going to use three feature selection methods in order to find the best accuracy at distinguishing faces which are: Relieff, Fisher Score, and Chi Square. . In the original project by Salem Alelyani he used the 10-fold cross validation and KNN Classifier to train and classify the images. However, because our department doesn’t have the Bioinformatics toolbox we are going to try to train and classify with liblinear.

Relieff: is a filter-based feature selection method that aims to maximized margin between classes while minimizing within-class distance. In other words, the weight of each feature is given based on how well it maximizes the margin between the sample and its nearest miss and minimizes the distance between the sample and its nearest hit.

Algorithm:
1. Initiate the weights vector to zero: \( w = 0 \).
2. for \( t = 1 . . . T \),
   (a) pick randomly an instance \( x \) from \( S \)
   (b) for \( i = 1 . . . N \),
      i. \( w_i = w_i + (x_i - \text{nearmiss}(x)_i)^2 - (x_i - \text{nearhit}(x)_i)^2 \)
3. the chosen feature set is \( \{ i \mid w_i > \tau \} \) where \( \tau \) is a threshold.
**Fisher Score:** is another affective filter-based feature selection method that assigns a weight to each feature based on Fisher criteria. Similar to ReliefF, Fisher Score assigns higher weight to features that maximize the distance between samples from different classes while the distance between samples from the same class is as small as possible.

**Fisher scoring algorithm:**

The partial derivatives of the likelihood for individual $j$ with respect to the parameters are

$$
\begin{align*}
\frac{\partial}{\partial \beta} L_j(\theta) &= -\frac{1}{2\sigma_j^2} \left( \frac{\partial}{\partial \beta} \sigma_j^2 \right) + \frac{1}{2} \left[ \frac{(y_j-\mu_j)^2}{\sigma_j^2} \left( \frac{\partial}{\partial \sigma_j^2} \sigma_j^2 \right) + \frac{2(y_j-\mu_j)}{\sigma_j^2} \left( \frac{\partial}{\partial \mu_j} \mu_j \right) \right] \\
\frac{\partial}{\partial \sigma^2} L_j(\theta) &= -\frac{1}{2\sigma_j^2} \left( \frac{\partial}{\partial \sigma^2} \sigma_j^2 \right) + \frac{1}{2} \left[ \frac{(y_j-\mu_j)^2}{\sigma_j^4} \left( \frac{\partial}{\partial \sigma^2} \sigma_j^2 \right) + \frac{2(y_j-\mu_j)}{\sigma_j^2} \left( \frac{\partial}{\partial \mu_j} \mu_j \right) \right]
\end{align*}
$$

where

$$
\begin{align*}
\frac{\partial}{\partial \beta} L_j(\theta) &= -\frac{1}{\sigma_j^2} \Sigma_j \beta + \frac{1}{\sigma_j^2} (y_j - \mu_j)^2 \Sigma_j \beta + \frac{1}{\sigma_j^2} U_j^T (y_j - \mu_j) \\
\frac{\partial}{\partial \sigma^2} L_j(\theta) &= -\frac{1}{2\sigma_j^2} + \frac{1}{2\sigma_j^4} (y_j - \mu_j)^2
\end{align*}
$$

Further manipulation on Equation (7) leads to

\[ X^2(x_i) = \sum [e(c, x_i) - o(c, x_i)] / e(c, x_i), \ c \in C \]

where the expected number of feature $x_i$ in $C$ is computed as: $e(c, x) = n_c (a_{xi} / N)$

Where $n_c$ is the number of proteins in subfamily $c$, $N$ is the total number of proteins and $a_{xi}$ is the number of proteins having feature $x$ at position $i$. 
III. Examples Illustrating the Main Steps of the Algorithm

Relieff Example:

\[ \text{[RANKED,WEIGHT]} = \text{relieff}(X,Y,K) \]

load fisheriris
[ranked,weight] = relieff(meas,species,10)
ranked =
    4     3     1     2
weight =
    0.1399    0.1226    0.3590    0.3754

Fisher Score Example:

\[ \text{[out]} = \text{fsFisher}(X,Y) \]

Output:
   \textbf{out}: A struct containing the following fields
   \textbf{W} - The distribution at each data point.
   \textbf{fList} - The list of features that are deemed useful.
   \textbf{prf} - This means that the smaller the feature weight is, the more useful it will be to the user.

Input:
   \textbf{X}: The features on current trunk, each column is a feature vector on all instances, and each row is a part of the instance.
   \textbf{Y}: The label of instances, in single column form: 1 2 3 4 5 ...

Chi Square Example:

function \[ \text{[chi, df]} = \text{chi2feature(feature, label)} \]
\%
\text{chi squared feature measure for binary class}
\%
\text{find class name}

IV. Implementation

% matrix D 100 x 3750
D = [tr_fea; ts_fea];
0.00% all class labels in training and testing. 
Y = [tr_label; ts_label];

% ReliefF feature selection method 
% Outputs the weight of the features and how they are ranked 
[RANKED, WEIGHT] = relieff(D, Y, 20);

% Fisher score feature Selection Method 
% Outputs the weight at each data point, also the 
% the flist of features that are useful, 
% and the prf of the small useful features 
[out] = fsFisher(D,Y);

% Chi Square Selection Method 
% outputs chi value and df 
% the chi value needs to be compared to the 
% chi squared table to see how important are 
% the features. 
[chi, df] = chi2feature(D, Y);

V. Test Results and Discussion
It is hard to say that my Results are accurate or closely accurate do to the complications faced in this project. The toolboxes needed for this project weren't available such as the Bioinformatics toolbox. Also do to the unsimilar matlab programs from one lab room to another, the results have a chance of being inclusive when it comes to executing Relieff function. I do not know the reason for this, but I do know that Relieff function is working fine.

If it all goes well, I put the extracted features by each of these feature selection methods into D2, D3, and D4 using the sort function. Which is another code I wasn't able to test, but theoretically should work since it is provided in the symbolic math toolbox. Therefore, D2 holds the feature list of Relieff, D3 holds the feature list of Fisher Score, and D4 holds the feature list of Chi Square. The feature lists assigned to their D variables are then run through liblinear, which should compute the accuracy of each selection method. Again, this was unable to test, since the matlab program in the Computer tutoring center wouldn't execute properly. Consequently, the accuracies should be close to 99%. This accuracy should be reached by using about 25 selected features. In the other hand, fisher and Chi Square use a larger number of selected features to reach an accuracy of 99%.

VI. Conclusion
The challenge was quite difficult, especially if you are sort of new to matlab. However, it was a neat experience to undertake. Much was learned through out the project. For example, there are many feature selection methods other than the ones used in this project like: sigma, HoG, Laplacian, and many others. That can achieve more or less the same
results like the methods used in this project. As for the results, Relieff should be the better Feature selection method of all three with an accuracy of 99% with about 20-25 selected features.

References


