Implemented Linear SVM with one-vs-all strategy in MatLab, to categorize images in one of 40 categories, with 86% accuracy.

Used images 1, 3, 4, 7 and 10 from each category as training set, and images 2, 5, 6, 8 and 9 for testing set, and then reversed the training and testing sets.

I've calculated α using MatLab's quadprog function.

```
 B = [Z'; zeros(199, 200)];

 H = (X*X').*(Z*Z');

alpha = quadprog(H + eye(200)*0.1, f, A, a, B, b);
```

X is the Matrix which contains all image vectors, and Z is the vector which specifies which class is being trained.

B changes with Z, as Z is the first row in the matrix and all other elements are 0.

Matrix H is $H_{ij} = z_i z_j x_i^t x_j$. This is represented in MatLab as : `H = (X*X').*(Z*Z');` .

From each obtained alpha, I have calculated W and W0 for each category, and stored them in arrays Ws and W0s, which have stored permanently on disk with their respective names as txt files.

For every new image X, I am calculating its predicted category using the following formula :

$$g(x) = w^t x + w_0$$

This is represented in MatLab as:

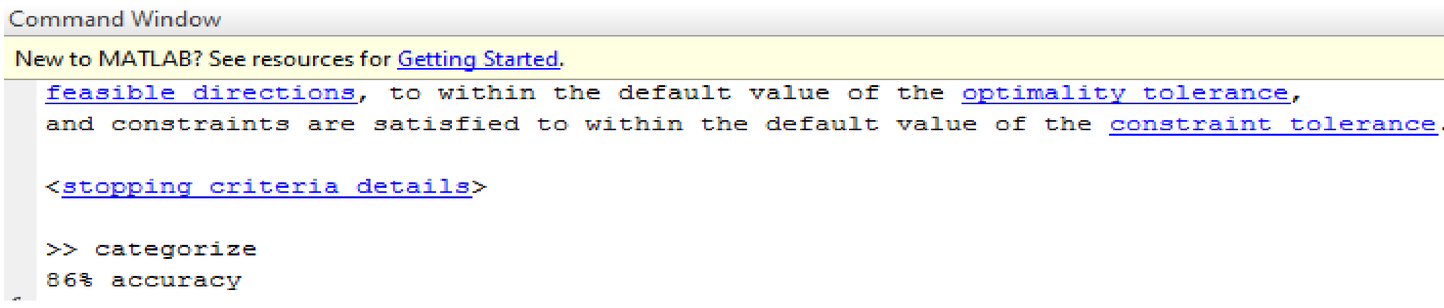
```
for j=1:40
     temp = dot(images(i,:), Ws(j,:)) + W0s(j);
     temps = [temps; temp];
end
```

I have taken the corresponding index of the maximum of these categories max(temps), and output it as the predicted class.

Writefiles.m calculates the W vector and W0 for each class, and writes them to their corresponding txt files for ease of access in future.

Categorize.m takes the files which are not in the training set, and predicts their category.

Here is a screenshot :



Command Window

New to MATLAB? See resources for Getting Started.

```
feasible directions, to within the default value of the optimality tolerance,
and constraints are satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

>> categorize
86% accuracy
```