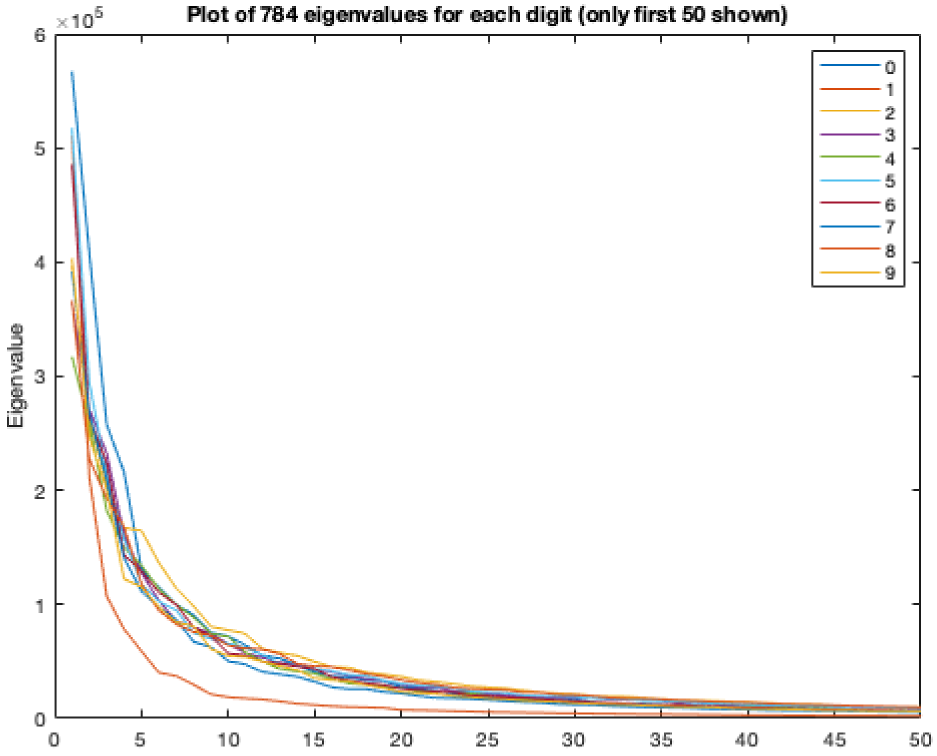
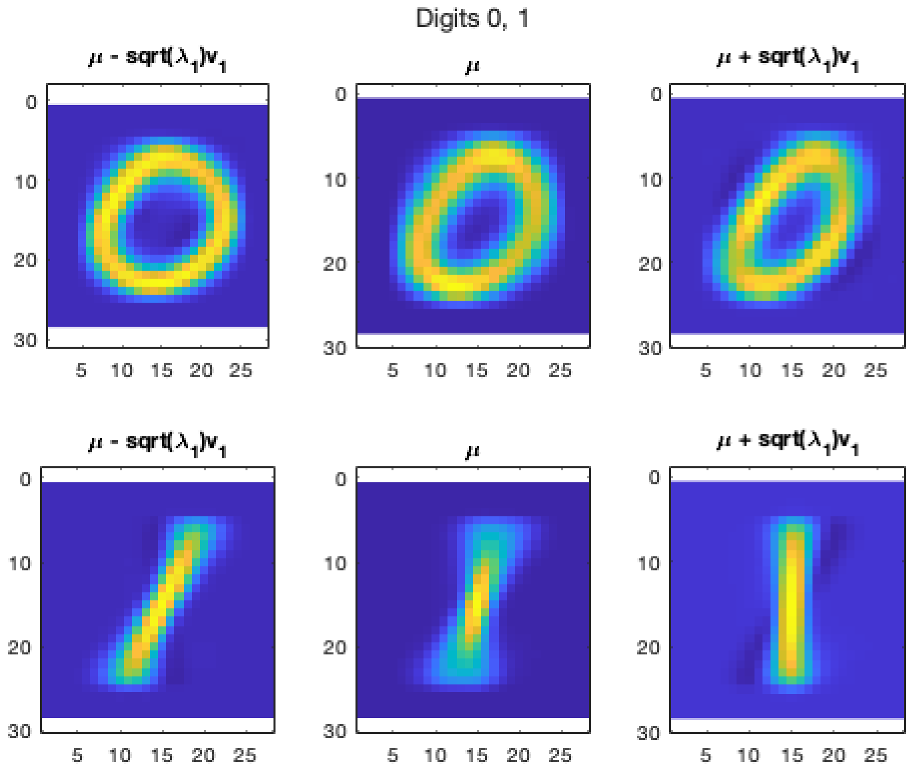
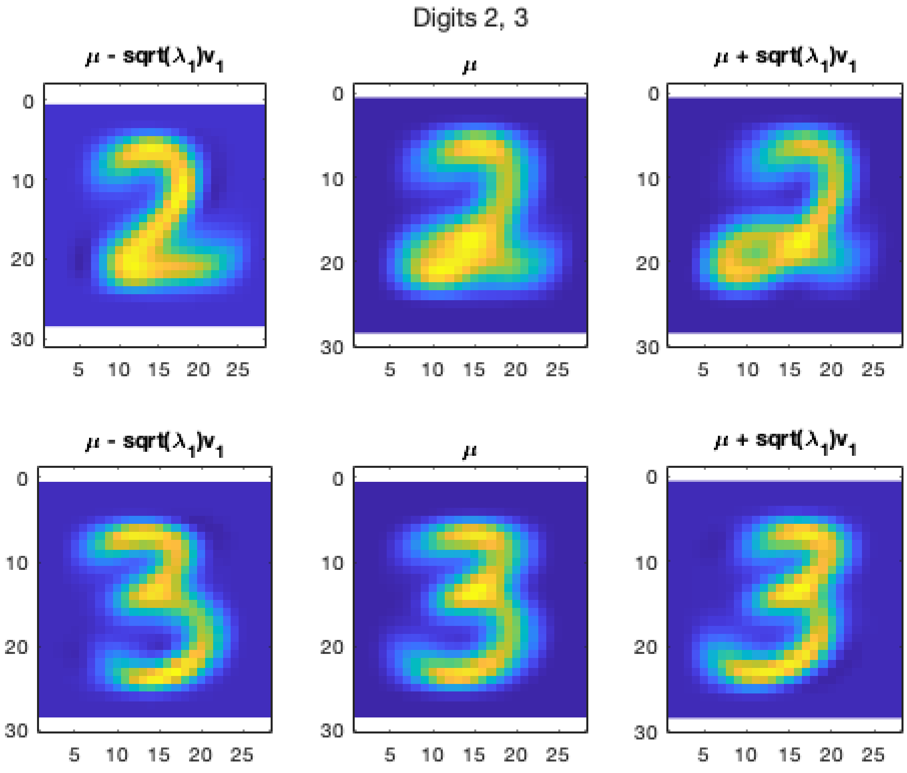
```matlab
clear;
close all;
S = load('../data/mnist.mat');

digits = double(S.digits_train);
digits = reshape(digits, 784, size(digits, 3));         % digits is now a 784x60000 matrix
labels = transpose(S.labels_train);                     % labels is a 1x60000 row

% Some matrices to store the results
Means = zeros(784, 10);
Covs = zeros(784, 784, 10);
Eigvecs = zeros(784, 10);
Eigvals = zeros(10);

for i = 0:9
    indices = find(labels == i);                        % extracting indices of current digit
    current_digit = digits(:, indices);
    N = size(current_digit, 2);                         % Number of samples
    mean = sum(current_digit, 2)/N;                     % 784x1 vector
    current_digit = current_digit - mean;
    C = current_digit * current_digit' / max(1, N-1);   % 784x784 covariance matrix

    [U, S] = eig(C);
    [D, ind] = sort(diag(S), 'descend');

    Means(:, i+1) = mean;
    Covs(:, :, i+1) = C;
    Eigvecs(:, i+1) = U(:, ind(1));
    Eigvals(i+1) = D(1);

    plot(D);
    hold on;
end
hold off;
xlim([0 50]);                       % Truncating rest of the values
legend('0', '1', '2', '3', '4', '5', '6', '7', '8', '9');
title('Plot of 784 eigenvalues for each digit (only first 50 shown)');
ylabel('Eigenvalue');

% Last part
for i = 0:9
    if mod(i, 2) == 0
        figure;
        t = tiledlayout(2, 3);
        title(t, ['Digits ', num2str(i), ', ', num2str(i+1)]);
    end

    nexttile;
    imagesc(reshape(Means(:, i+1) - sqrt(Eigvals(i+1))*Eigvecs(:, i+1), 28, 28));
    axis equal;
    title('\mu - sqrt(\lambda_1)v_1');

    nexttile;
    imagesc(reshape(Means(:, i+1), 28, 28));
    axis equal;
    title('\mu');

    nexttile;
    imagesc(reshape(Means(:, i+1) + sqrt(Eigvals(i+1))*Eigvecs(:, i+1), 28, 28));
    axis equal;
    title('\mu + sqrt(\lambda_1)v_1');
end
```

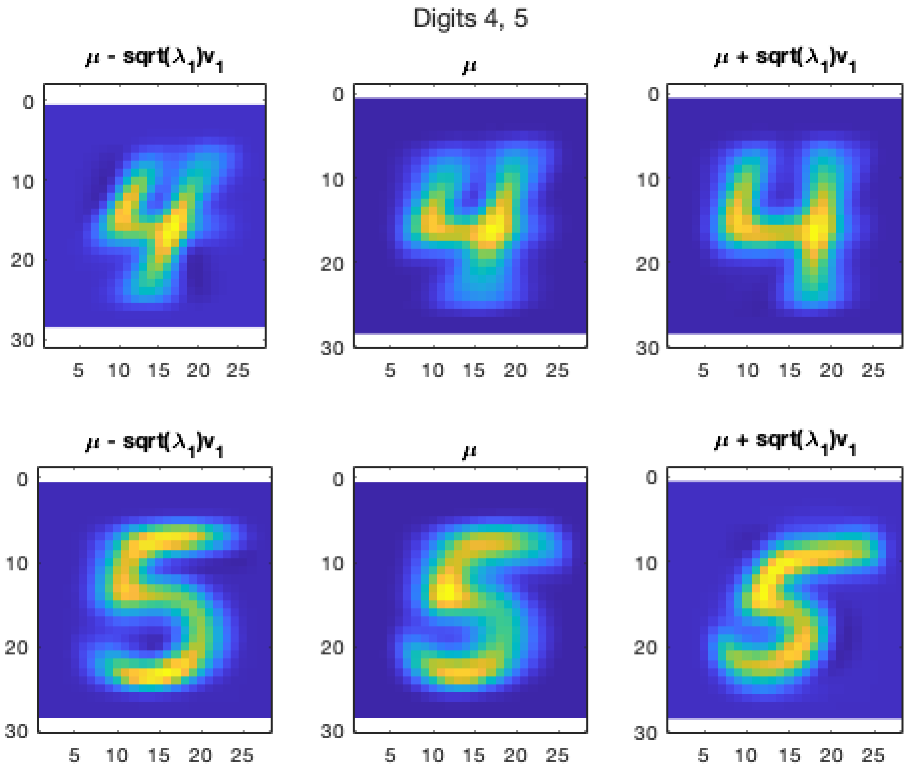


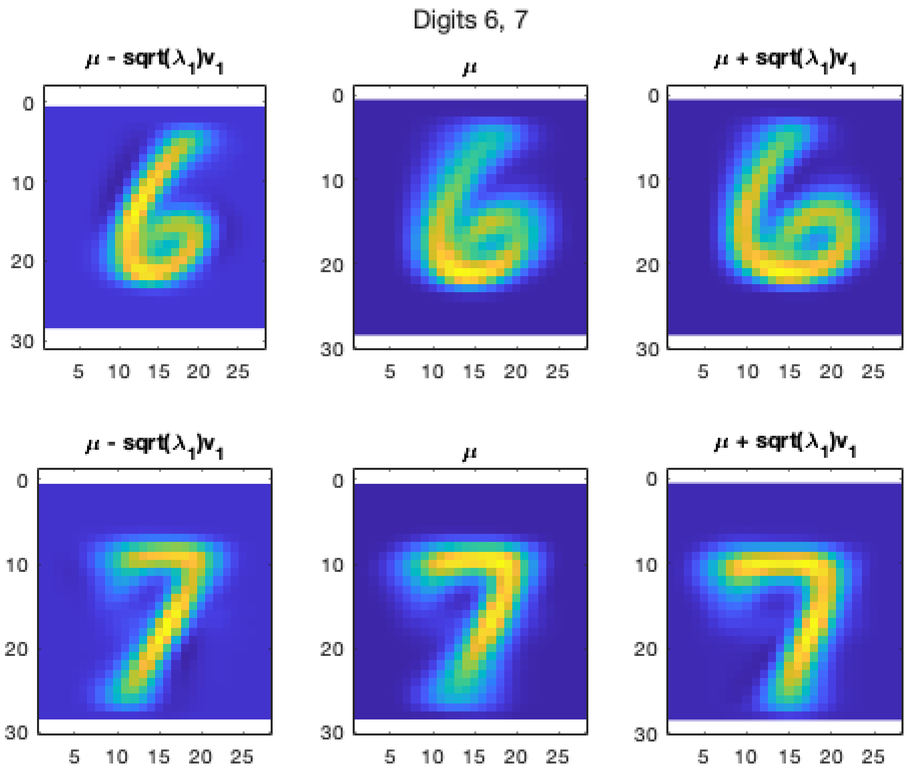
Plot of 784 eigenvalues for each digit (only first 50 shown)
×10^5
Eigenvalue
6
5
4
3
2
1
0
0
5
10
15
20
25
30
35
40
45
50
0
1
2
3
4
5
6
7
8
9

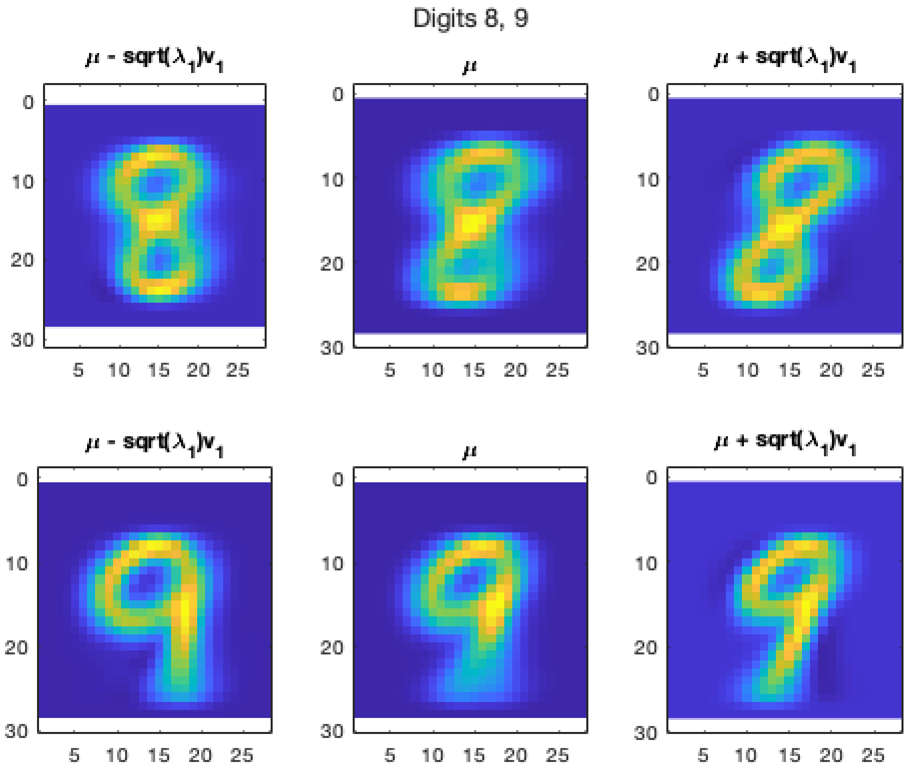


Digits 0, 1
μ - sqrt(λ_1)v_1
μ
μ + sqrt(λ_1)v_1
μ - sqrt(λ_1)v_1
μ
μ + sqrt(λ_1)v_1
0
10
20
30
5
10
15
20
25


Digits 2, 3
μ - sqrt(λ1)v1
μ
μ + sqrt(λ1)v1
0
10
20
30
5
10
15
20
25


Digits 4, 5
μ - sqrt(λ1)v1
μ
μ + sqrt(λ1)v1
0
10
20
30
5
10
15
20
25





Published with MATLAB® R2020a