```
addpath(genpath('data'));
addpath(genpath('Matlab'));

%%

[multiIm, annotationIm] = loadMulti('multispectral_day01.mat','annotation_day01.png');

[meanThresholds,idx] = computeMeanThresholds(multiIm, annotationIm);

[fatPix, fatR, fatC] = getPix(multiIm, annotationIm(:,:,2));
[meatPix, meatR, meatC] = getPix(multiIm, annotationIm(:,:,3));

%%

figure (1)
plot(mean(meatPix), 'b');
hold on
plot(mean(fatPix), 'r');
hold on
plot(meanThresholds,'g');
legend('Fat pixels','Meat pixels','Thresholds');

%%

figure(2)
plot(showHistograms(multiIm,annotationIm(:,:,3),idx,false),'b');
hold on
plot(showHistograms(multiIm,annotationIm(:,:,2),idx,false),'r');
hold on
xline(mean(meatPix(:,idx)),'b');
hold on
xline(mean(fatPix(:,idx)),'r');
hold on
xline(meanThresholds(idx),'k');
xlim([5 65]);
legend('Meat','Fat');

%%

% For each spectral layer, the number of fat and meat pixels on the 'wrong'
% side of the threshold i counted.

errorRate = zeros(1,19);

for l = 1:18
    for p = 1:length(fatPix)
        if fatPix(p,l) < meanThresholds(l)
            errorRate(l) = errorRate(l) + 1;
        end
    end
    for p = 1:length(meatPix)
        if meatPix(p,l) > meanThresholds(l)
            errorRate(l) = errorRate(l) + 1;
        end
    end
end

figure(2)
plot(errorRate);

%%
```

```
% Combining annotations to get image of background.

background = sum(annotationIm,3);

%%

% Converting from int8 to double.

multiImDouble = double(multiIm);

%%

meatClass = zeros(514);

% Classifying every meat-pixel and giving it value 1.

for i = 1:514
    for j = 1:514
        if multiImDouble(i,j,idx) < meanThresholds(idx) && background(i,j) == 1
            meatClass(i,j) = 1;
        end
    end
end

figure(3);
imshow(meatClass);
title('Meat-pixels classified and colored white with simple model');

%%
pFat = 0.3;
pMeat = 0.7;

[Sf_fat,Sf_meat] = computeSFunctions(multiIm,annotationIm,pFat,pMeat);

%%

% Computing Sfat and Smeat for each pixel.

Sfat = zeros(514);
Smeat = zeros(514);

for i = 1:514
    for j = 1:514
        Sfat(i,j) = Sf_fat(multiImDouble(i,j,:));
        Smeat(i,j) = Sf_meat(multiImDouble(i,j,:));
    end
end

%%

% Computing difference. If Sdif > 1 the probability for meat is largest.

Sdif = Smeat./Sfat;

%%

% Coloring in every meat pixel.

advClassDay1 = zeros(514);

for i = 1:514
    for j = 1:514
```

```
        if Sdif(i,j) > 1 && background(i,j) == 1
            advClassDay1(i,j) = 1;
        end
    end
end

%%

figure(4);
imshow(advClassDay1);
title('Meat-pixels classified and colored white with advanced model');

%%

% Counting the errors made on day 1 using advanced method.

absErrorDay1 = 0;

for i = 1:514
    for j = 1:514
        if advClassDay1(i,j) ~= annotationIm(i,j,3) && annotationIm(i,j,3) == 1
            absErrorDay1 = absErrorDay1 + 1;
        end
    end
end

%%

days = [1,6,13,20,28];
absErrorsAdv = zeros(1,5);
absErrorsSimple = zeros(1,5);
numPixels = zeros(1,5);

for k = 1:5
    [multiIm, annotationIm] = loadMulti(strcat('multispectral_day',sprintf('%02d',days↙
(k)),'.mat'),strcat('annotation_day',sprintf('%02d',days(k)),'.png'));

    multiImDouble = double(multiIm);

    background = sum(annotationIm,3);

    simpleClass = zeros(514);

    % Classifying using simple model. 1 = meat.

    for i = 1:514
        for j = 1:514
            if multiImDouble(i,j,idx) < meanThresholds(idx) && background(i,j) == 1
                simpleClass(i,j) = 1;
            end
        end
    end

    figure(5)
    subplot(2,3,k);
    imshow(simpleClass);
    title(strcat('Day',{' '}, int2str(days(k))));
    sgtitle('Classification using simple model');

    % Computing S-values for every pixel.

    Sfat = zeros(514);
```

```
    Smeat = zeros(514);

    for i = 1:514
        for j = 1:514
            if background(i,j) == 1
                Sfat(i,j) = Sf_fat(multiImDouble(i,j,:));
                Smeat(i,j) = Sf_meat(multiImDouble(i,j,:));
            end
        end
    end

    Sdif = Smeat./Sfat;

    % Classifying pixels using advanced model. 1 = meat.

    advClass = zeros(514);

    for i = 1:514
        for j = 1:514
            if Sdif(i,j) > 1 && background(i,j) == 1
                advClass(i,j) = 1;
            end
        end
    end

    figure(6)
    subplot(2,3,k);
    imshow(advClass);
    title(strcat('Day',{' '}, int2str(days(k))));
    sgtitle('Classification using advanced model');

    for i = 1:514
        for j = 1:514
            if annotationIm(i,j,3) == 1 || annotationIm(i,j,2) == 1
                numPixels(k) = numPixels(k) + 1;
            end
            % Counting errors using simple model.
            if (annotationIm(i,j,3) == 1 && simpleClass(i,j) == 0) || (annotationIm(i,↙
j,2) == 1 && simpleClass(i,j) == 1)
                absErrorsSimple(k) = absErrorsSimple(k) + 1;
            end
            % Counting errors using advanced model.
            if (annotationIm(i,j,3) == 1 && advClass(i,j) == 0) || (annotationIm(i,j,↙
2) == 1 && advClass(i,j) == 1)
                absErrorsAdv(k) = absErrorsAdv(k) + 1;
            end
        end
    end
end

%%

errorRateAdv = absErrorsAdv./numPixels;
errorRateSimple = absErrorsSimple./numPixels;

%%

figure(7)
plot(days,errorRateAdv,'b');
hold on
plot(days,errorRateSimple,'r');
```

```
%%

pFat = 0.3;
pMeat = 0.7;

days = [1,6,13,20,28];
absErrorsMeatAdv = zeros(5,5);
absErrorsFatAdv = zeros(5,5);

absErrorsMeatSimple = zeros(5,5);
absErrorsFatSimple = zeros(5,5);

errorRateMeatAdv = zeros(5,5);
errorRateFatAdv = zeros(5,5);
errorRateAdv = zeros(5,5);

errorRateMeatSimple = zeros(5,5);
errorRateFatSimple = zeros(5,5);
errorRateSimple = zeros(5,5);

numPixelsMeat = zeros(1,5);
numPixelsFat = zeros(1,5);

for d = 1:5 % Looping through each day for training.
    [multiIm, annotationIm] = loadMulti(strcat('multispectral_day',sprintf('%02d',days↙
(d)),'.mat'),strcat('annotation_day',sprintf('%02d',days(d)),'.png'));

    [Sf_fat,Sf_meat] = computeSFunctions(multiIm,annotationIm,pFat,pMeat);

    [meanThresholds,idx] = computeMeanThresholds(multiIm,annotationIm);

    for k = 1:5 % Looping through every other day for classification.
        if k ~= d % Counting errors during classification of images from other days.
            [multiIm, annotationIm] = loadMulti(strcat('multispectral_day',sprintf('%↙
02d',days(k)),'.mat'),strcat('annotation_day',sprintf('%02d',days(k)),'.png'));

            multiImDouble = double(multiIm);

            background = sum(annotationIm,3);

            numPixelsMeat(k) = nnz(annotationIm(:,:,3));
            numPixelsFat(k) = nnz(annotationIm(:,:,2));

            simpleClass = zeros(514);

            % Classifying using simple model. 1 = meat.

            for i = 1:514
                for j = 1:514
                    if multiImDouble(i,j,idx) < meanThresholds(idx) && background(i,j)↙
== 1
                        simpleClass(i,j) = 1;
                    end
                end
            end

            % Computing S-values for every pixel.

            Sfat = zeros(514);
            Smeat = zeros(514);

            for i = 1:514
```

```
                for j = 1:514
                    if background(i,j) == 1
                        Sfat(i,j) = Sf_fat(multiImDouble(i,j,:));
                        Smeat(i,j) = Sf_meat(multiImDouble(i,j,:));
                    end
                end
            end

            Sdif = Smeat./Sfat;

            % Classifying pixels using advanced model. 1 = meat.

            advClass = zeros(514);

            for i = 1:514
                for j = 1:514
                    if Sdif(i,j) > 1 && background(i,j) == 1
                        advClass(i,j) = 1;
                    end
                end
            end

%             for i = 1:514
%                 for j = 1:514
%                     if (annotationIm(i,j,3) == 1 && simpleClass(i,j) == 0)
%                         absErrorsMeatSimple(d,k) = absErrorsMeatSimple(d,k) + 1;
%                     elseif (annotationIm(i,j,2) == 1 && simpleClass(i,j) == 1)
%                         absErrorsFatSimple(d,k) = absErrorsFatSimple(d,k) + 1;
%                     end
%                     if (annotationIm(i,j,3) == 1 && advClass(i,j) == 0)
%                         absErrorsMeatAdv(d,k) = absErrorsMeatAdv(d,k) + 1;
%                     elseif (annotationIm(i,j,2) == 1 && advClass(i,j) == 1)
%                         absErrorsFatAdv(d,k) = absErrorsFatAdv(d,k) + 1;
%                     end
%                 end
%             end

            misClassMeatSimple = getPix(annotationIm(:,:,3),~simpleClass);
            misClassFatSimple = getPix(annotationIm(:,:,2),simpleClass);

            misClassMeatAdv = getPix(annotationIm(:,:,3),~advClass);
            misClassFatAdv = getPix(annotationIm(:,:,2),advClass);

            absErrorsMeatSimple(d,k) = nnz(misClassMeatSimple);
            absErrorsFatSimple(d,k) = nnz(misClassFatSimple);

            absErrorsMeatAdv(d,k) = nnz(misClassMeatAdv);
            absErrorsFatAdv(d,k) = nnz(misClassFatAdv);

            errorRateMeatAdv(d,k) = absErrorsMeatAdv(d,k)/numPixelsMeat(k);
            errorRateFatAdv(d,k) = absErrorsFatAdv(d,k)/numPixelsFat(k);
            errorRateAdv(d,k) = (absErrorsMeatAdv(d,k)+absErrorsFatAdv(d,k))/↙
(numPixelsFat(k)+numPixelsMeat(k));

            errorRateMeatSimple(d,k) = absErrorsMeatSimple(d,k)/numPixelsMeat(k);
            errorRateFatSimple(d,k) = absErrorsFatSimple(d,k)/numPixelsFat(k);
            errorRateSimple(d,k) = (absErrorsMeatSimple(d,k)+absErrorsFatSimple(d,k))/↙
(numPixelsFat(k)+numPixelsMeat(k));

        end
    end
end
```

```
%%

for i = 1:5
    figure(8)
    subplot(1,5,i);
    bar(days,errorRateAdv(i,:),'b');
    hold on
    bar(days,errorRateSimple(i,:),'r');
    title(strcat('Trained on day',{' '}, int2str(days(i))));
    legend('Advanced model','Simple model')
end

%%

for i = 1:5
    figure (10)
    subplot(1,5,i);
    y = [errorRateAdv(i,:);errorRateSimple(i,:)]';
    bar(days,y)
    ylim([0.0 0.15])
    title(strcat('Trained on day',{' '}, int2str(days(i))));
end
legend('Advanced model','Simple model')

%%

meanErrorRateSimple = mean(errorRateSimple,2);
meanErrorRateAdv = mean(errorRateAdv,2);

%%

% Used for comparing with and wihtout prior.

meanErrorRateAdv2 = mean(errorRateAdv,2);

%%

meanErrorRateAdvDif = meanErrorRateAdv-meanErrorRateAdv2;

%%

figure(13)
bar(days,meanErrorRateAdv);
hold on
bar(days,meanErrorRateAdvDif);
legend('Advanced model with prior','Difference without prior')

%%

figure(11);
bar(days,[meanErrorRateAdv,meanErrorRateSimple]);
legend('Advanced model','Simple model')

%%
```