

```

path='/Users/utkarsh/Downloads/distributed/';

labels = zeros(1,36);
labels(19:36)=1;

a=dir([path '/*.ppm']);
numimages=size(a,1);

% Feature Extraction
feat1 = zeros(1,numimages);
feat2 = zeros(1,numimages);

str = strcat('%simg_00',num2str(1),'.ppm');

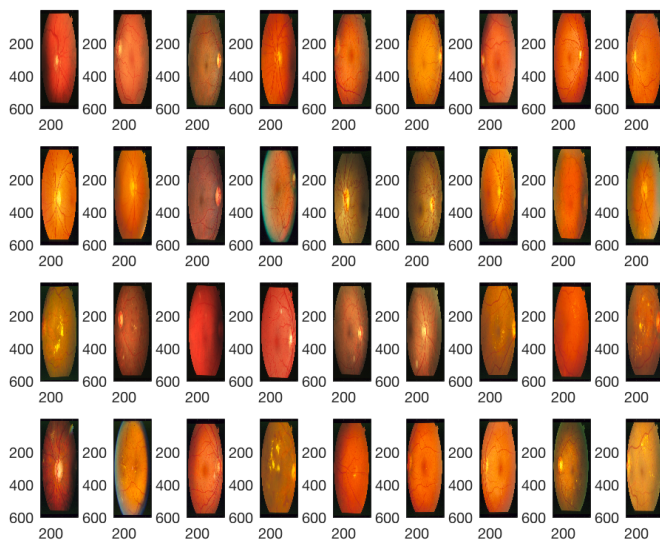
img = double(imread(sprintf(str, path)));

img = img(:,:,2);

% Make mask which will later be used to remove the edges of the eye
mask = img>(0.1*max(max(img)));
se = strel('disk',60,8);
mask = imerode(mask,se);

figure
for j = 1:numimages
    if j<=9
        str = strcat('%simg_00',num2str(j),'.ppm');
    end
    if j>9
        str = strcat('%simg_0',num2str(j),'.ppm');
    end
    if j>99
        str = strcat('%simg_',num2str(j),'.ppm');
    end
    img = imread(sprintf(str, path));
    subplot(4,9,j);
    image(img);
end

```



```

figure
for j = 1:numimages
    if j<=9
        str = strcat('%simg_00',num2str(j),'.ppm');
    end

```

```

if j>9
    str = strcat('%simg_0',num2str(j),'.ppm');
end
if j>99
    str = strcat('%simg_',num2str(j),'.ppm');
end
img = double(imread(sprintf(str, path)));

% Take only the green channel
img = img(:,:,2);

% Subtract median filtered image to remove intensity gradient
img = img - medfilt2(img,[5 5]);
img = img - min(min(img));
img = (img./max(max(img)))*255;

% Windowing function
a = 0.61*max(max(img));
b = 0.95*max(max(img));

img(img<a) = 0;
img(img>b) = 255;
img(img>=a & img<=b) = (255/(b-a))*img(img>=a & img<=b) - 255*a/(b-a);

img(img<a) = 0;
img(img>a) = 1;

% Dilate spots to merge spots close to each other
se = strel('disk',20);
img = imdilate(img,se);

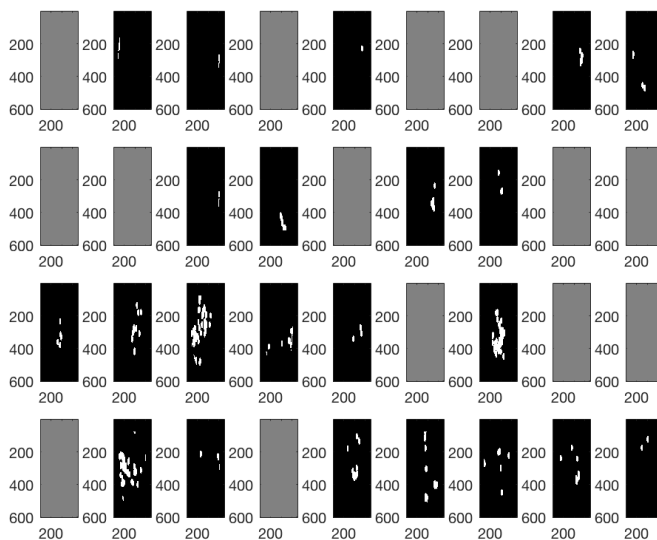
% Multiply image by mask to remove the outlines of the eye
img = mask .* img;

subplot(4,9,j);
imagesc(img);
colormap(gray);

% Count number of connected components
cc = bwconncomp(img);
feat1(1,j) = cc.NumObjects;

% Count total area of white spots
feat2(1,j) = sum(sum(img));
end

```



```

% Threshold to get feature values
feat1(feat1<=1) = 0;
feat1(feat1>1) = 1;

threshold = 2500;

feat2(feat2<=threshold) = 0;
feat2(feat2>threshold) = 1;

feat = [feat1', feat2'];

% 6 fold cross validation using KNN

folds1= repmat([1:6], 1,3);
folds1 = folds1(randperm(18));
folds2 = repmat([1:6], 1,3);
folds2 = folds2(randperm(18));

folds = [folds1, folds2];

k=7;

pred = zeros(1, numimages);

for j=1:6
    train = feat(folds~=j, :);
    test = feat(folds==j, :);
    labels_train = labels(folds ~= j);

    % Code for scaling
    % Commented out since both features have a value of either 0 or 1
    %nfeat = size(train, 2);
    %for n=1:nfeat
    %    mn_train = mean(train(:,n));
    %    sd_train = std(train(:,n));
    %    train(:,n) = (train(:,n)-mn_train)/sd_train;
    %    test(:,n) = (test(:,n)-mn_train)/sd_train;
    %end

    ntest = size(test, 1);
    ntrain = size(train, 1);
    pred_test = zeros(1, ntest);

    for i=1:ntest
        dist_from_train = sqrt(sum((ones(ntrain,1)*test(i,:)-train).^2, 2));
        [reord, ord] = sort(dist_from_train);
        knn = labels_train(ord(1:k));
        p_g1 = mean(knn == 0);
        p_g2 = mean(knn == 1);
        if (p_g2<p_g1)
            pred_test(i)=0;
        elseif (p_g1<p_g2)
            pred_test(i)=1;
        else
            pred_test(i) = randperm(2,1)-1;
        end
    end
    pred(folds == j) = pred_test;
end

disp("The accuracy for healthy retina's using KNN is "+num2str(mean(pred(labels ==0)==0)));

```

The accuracy for healthy retina's using KNN is 0.83333

```
disp("The accuracy for unhealthy retina's using KNN is "+num2str(mean(pred(labels ==1)==1)));
```

The accuracy for unhealthy retina's using KNN is 0.72222

```

% 6 fold cross validation using Logistic Regression

numfolds = 6;

labels = labels';

```

```

folds1= repmat([1:numfolds], 1,18/numfolds);
folds1 = folds1(randperm(18));
folds2 = repmat([1:numfolds], 1,18/numfolds);
folds2 = folds2(randperm(18));

folds = [folds1, folds2];

pred = zeros(1, numimages);

for j=1:numfolds

    train = feat(folds~=j, :);
    test = feat(folds==j, :);
    labels_train = labels(folds ~= j);

    % Code for scaling
    % Commented out since both features have a value of either 0 or 1
    %nfeat = size(train, 2);
    %for n=1:nfeat
    %    mn_train = mean(train(:,n));
    %    sd_train = std(train(:,n));
    %    train(:,n) = (train(:,n)-mn_train)/sd_train;
    %    test(:,n) = (test(:,n)-mn_train)/sd_train;
    %end

    ntest = size(test, 1);
    ntrain = size(train, 1);
    pred_test = zeros(1, ntest);

    beta = glmfit(train, labels_train, 'binomial', 'link', 'logit');

    xb = [ones(size(test,1), 1), test]*beta;
    prob_test = exp(xb)./(1+exp(xb));
    pred_test = 1*prob_test>.5;

    pred(folds == j) = pred_test;
end

disp("The accuracy for healthy retina's using Logistic Regression is "+num2str(mean(pred(labels ==0)==0)));

```

The accuracy for healthy retina's using Logistic Regression is 0.72222

```

disp("The accuracy for unhealthy retina's using Logistic Regression is "+num2str(mean(pred(labels ==1)==1)));

```

The accuracy for unhealthy retina's using Logistic Regression is 0.72222