clc

clear

digitDatasetPath = fullfile(matlabroot,'toolbox','nnet','nndemos', ...

'nndatasets','volnteerb1');

imds = imageDatastore(digitDatasetPath, ...

'IncludeSubfolders',true, ...

'LabelSource','foldernames');

%deepNetworkDesigner

numTrainFiles = 16; %number of classes (no. of folder created inside volunteer b)

labelCount = countEachLabel(imds);

numClasses = height(labelCount);

net = alexnet;

%net.Layers

%% Create training and validation sets

%net = inceptionv3;net.layers%analyzeNetwork(net);

[imdsTrain, imdsValidation] = splitEachLabel(imds, 0.5, 'randomize');

inputSize = net.Layers(1).InputSize;

layersTransfer = net.Layers(1:end-3);

numClasses = numel(categories(imdsTrain.Labels));

numTrainImages = numel(imdsTrain.Labels);

figure

for i=1:16

subplot(4,4,i)

I= readimage(imdsTrain,i);

imshow(I)

title('training')

end

layers = [

layersTransfer

fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRateFactor',20)

softmaxLayer

classificationLayer];

pixelRange = [-30 30];

imageAugmenter = imageDataAugmenter( ...

'RandXReflection',true, ...

'RandXTranslation',pixelRange, ...

'RandYTranslation',pixelRange);

augimdsTrain = augmentedImageDatastore(inputSize(1:2),imdsTrain, ...

'DataAugmentation',imageAugmenter);

YValidation = imdsValidation.Labels;

augimdsValidation = augmentedImageDatastore(inputSize(1:2),imdsValidation);

%miniBatchSize = 10;

%valFrequency = floor(numel(augimdsTrain.Files)/miniBatchSize);

options = trainingOptions('sgdm', ...

'MiniBatchSize',16, ...%10

'MaxEpochs',2, ... %20

'InitialLearnRate',1e-4, ...

'Shuffle','every-epoch', ...

'ValidationData',{augimdsValidation,YValidation}, ...

'ValidationFrequency',10, ...

'Verbose',true, ...

'Plots','training-progress')

%'OutputFcn',@(info)savetrainingplot(info));

% validationFrequency = floor(numel(augimdsValidation)/128);

% options1 = trainingOptions('sgdm', ...

% 'InitialLearnRate',1e-4, ...

% 'Shuffle','every-epoch', ...

% 'MaxEpochs',6, ...

% 'MiniBatchSize',128, ...

% 'VerboseFrequency',30, ...

% 'ValidationData',{augimdsValidation,YValidation}, ...

% 'ValidationFrequency',30, ...

% 'Verbose',true, ...

% 'Plots','training-progress');

rng('default');

netTransfer = trainNetwork(augimdsTrain,layers,options);

[YPred,scores] = classify(netTransfer,augimdsValidation);

idx1 = randperm(numel(imdsValidation.Files),16);%16

figure(2)

for i = 1:16%1:16

subplot(4,4,i)

I = readimage(imdsValidation,idx1(i));

imshow(I)

label = YPred(idx1(i));

title(string(label));

end;

%Top 1 Accuracy:

YValidation = imdsValidation.Labels;

Top1accuracy = mean(YPred == YValidation)

accuracy = sum(YPred == YValidation)/numel(YValidation)

%Top 5 Accuracy:

YValidation = imdsValidation.Labels;

[n,m] = size(scores);

idx = zeros(m,n);

for i=1:n

[~,idx(:,i)] = sort(scores(i,:),'descend');

end

%idx = idx(5:-1:1);

idx = idx(1:16,:); %idx = idx(1:3,:); <=======

top5Classes = netTransfer.Layers(end).ClassNames(idx);

scoresTop = scores(idx);

figure

barh(mean(scoresTop'))

xlim([0 1])

title('Predictions Accuracy ')

xlabel('Probability')

yticklabels(top5Classes)%

top5count = 0;

for i = 1:n

top5count = top5count + sum(YValidation(i,1) == top5Classes(:,i));

end

top5Accuracy = top5count/n

%idx = idx(1:10:-1:1);

scoresTop = scores(idx);

figure

barh(scoresTop)

xlim([0 1])

title('Acuracy Predictions for 16 classes')%title('Top 5 Predictions')<====

xlabel('Probability')

yticklabels(top5Classes)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% predicted\_labels=YPred

% actual\_labels=imds.Labels;

conf=confusionmat(YValidation,YPred)

figure;

plotconfusion(YValidation,YPred)

title('Confusion Matrix: AlexNet');

% figure;

% plotconfusion(actual\_labels,predicted\_labels')

% title('Confusion Matrix');

% %ROC CURVE

% test\_labels=double(nominal(imds.Labels));

%

% % ROC Curve - Our target class is the first class in this scenario

% [fp\_rate,tp\_rate,T,AUC]=perfcurve(test\_labels,posterior(:,1),1);

% %figure;

% %plot(fp\_rate,tp\_rate,'b-');

% %grid on;

% %xlabel('False Positive Rate');

% %ylabel('Detection Rate');

% % Area under the ROC curve value

% AUC

% %evaluation

% %Evaluate(YValidation,YPred)

% ACTUAL=actual\_labels;

% PREDICTED=predicted\_labels';

% idx = (ACTUAL()==total\_split.Label(1));

% %disp(idx)

% p = length(ACTUAL(idx));

% n = length(ACTUAL(~idx));

% N = p+n;

% tp = sum(ACTUAL(idx)==PREDICTED(idx));

% tn = sum(ACTUAL(~idx)==PREDICTED(~idx));

% fp = n-tn;

% fn = p-tp;

%

% tp\_rate = tp/p;

% tn\_rate = tn/n;

%

% accuracy = (tp+tn)/N;

% sensitivity = tp\_rate;

% specificity = tn\_rate;

% precision = tp/(tp+fp);

% recall = sensitivity;

% f\_measure = 2\*((precision\*recall)/(precision + recall));

% gmean = sqrt(tp\_rate\*tn\_rate);

% t=[AUC,accuracy,sensitivity,specificity,precision,recall,f\_measure,gmean];

% x={t,path,optimizer,augmentation,numfold};